LNCIS
Lecture Notes in Control and Information Sciences
401

Graziano Chesi
Koichi Hashimoto (Eds.)

Visual Servoing via Advanced Numerical Methods

Springer

Lecture Notes
in Control and Information Sciences 401

Editors: M. Thoma, F. Allgöwer, M. Morari

Graziano Chesi and Koichi Hashimoto (Eds.)

Visual Servoing via Advanced Numerical Methods

Springer

Editors

Prof. Graziano Chesi

University of Hong Kong
Department of Electrical
and Electronic Engineering
Pokfulam Road
Hong Kong
Chow Yei Ching Building
China, People's Republic
E-mail: chesi@eee.hku.hk

Prof. Koichi Hashimoto

Tohoku University
Dept. System Information Sciences
6-6-01 Aramaki-Aza Aoba
Sendai
Aoba-ku
980-8579 Japan
E-mail: koichi@ic.is.tohoku.ac.jp

ISBN 978-1-84996-088-5 e-ISBN 978-1-84996-089-2

DOI 10.1007/978-1-84996-089-2

Lecture Notes in Control and Information Sciences ISSN 0170-8643

Library of Congress Control Number: Applied for

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

5 4 3 2 1 0

springer.com

To my family Shing Chee, Isabella and Sofia (GC).

To my family Ritsuko, Riho, Daisuke, Kanta, Teppei and Shogo (KH).

This book is dedicated in memory of Dr. Ryosuke Mori (1972–2008).

Preface

Robots able to imitate human beings have been at the core of stories of science fiction as well as dreams of inventors for a long time. Among the various skills that Mother Nature has provided us with and that often go forgotten, the ability of sight is certainly one of the most important. Perhaps inspired by tales of Isaac Asimov, comics and cartoons, and surely helped by the progress of electronics in recent decades, researchers have progressively made the dream of creating robots able to move and operate by exploiting artificial vision a concrete reality.

Technically speaking, we would say that these robots position themselves and their end-effectors by using the view provided by some artificial eyes as feedback information. Indeed, the artificial eyes are visual sensors such as cameras that have the function to acquire an image of the environment. Such an image describes if and how the robot is moving toward the goal and hence constitutes feedback information. This procedure is known in robotics with the term *visual servoing*, and it is nothing else than an imitation of the intrinsic mechanism that allows human beings to realize daily tasks such as reaching the door of the house or grasping a cup of coffee.

Hence, there is no need to say that visual servoing is one of the hottest areas of robotics. Indeed, visual servoing can be, and is, exploited in numerous and various applications, such as industry, surveillance, transportation, exploration, surgery, and the replacement of human beings in dangerous environments. This book aims to provide a collection of the latest advances in visual servoing, which have been achieved through the development of dedicated numerical methods that the recent progress in video devices, computer hardware, and optimization techniques, have made possible.

Organization of the Book

The book is organized in three main parts, reflecting three main classes of issues in visual servoing.

The first part, *Vision*, comprises contributions whose main focus is on vision issues that characterize visual servoing tasks. Specifically, Mariottini *et al.* introduce visual servoing via catadioptric stereo with planar mirrors. Swensen *et al.* analyze convergence properties of featureless visual servoing. Namiki *et al.* propose a high-speed vision system for grasping and manipulation. Hager considers human-machine cooperation via vision-based motion constraints. Collewet and Marchand propose luminance as a new feature for visual servoing. Bachta *et al.* address vision issues in beating heart surgery.

The second part, *Estimation and Path-planning*, collects contributions whose main focus is on estimation issues and path-planning strategies. Specifically, Mezouar addresses trajectory planning via variational calculous. Malis *et al.* consider the estimation of homography dynamics. Chesi *et al.* investigate the effect of image measurement errors on the positioning error. Danes *et al.* analyze visual servoing through rational systems and linear matrix inequalities (LMIs). Kazemi *et al.* present a review of path-planning methods. Dani and Dixon consider structure and motion estimation with single camera. Tahri *et al.* address visual servoing and pose estimation with cameras obeying the unified model.

Lastly, the third part, *Control*, focuses on control issues of visual servoing tasks. Specifically, Ma and Hutchinson present gradient projection methods for taking into account constraints in image-based visual servoing (IBVS). Tarbouriech and Soueres address multi-constraint satisfaction in IBVS via LMIs. Hadj-Abdelkader *et al.* present a control scheme with central cameras. Lamiraux and Lefebvre describe deformation-based trajectory control for nonholonomic robots. Belo *et al.* consider eye-in-hand unicycle-like robots via position-based visual servoing and IBVS. Iwatani *et al.* address occlusion avoidance in unmanned helicopter control. Allibert *et al.* propose the use of nonlinear predictive control in visual servoing.

Acknowledgements

We thank the authors of the chapters for their precious contributions that have made possible to realize this great collection of advanced numerical methods for visual servoing. And we also thank the Springer Editors for their patience and suggestions that have helped us in the difficult task of editing a book.

This book was typeset by the authors using LaTeX.

Hong Kong
Sendai
September 2009

Graziano Chesi
Koichi Hashimoto

Contents

Part I: Vision

1 Catadioptric Stereo with Planar Mirrors: Multiple-view Geometry and Camera Localization 3
Gian Luca Mariottini, Stefano Scheggi, Fabio Morbidi, Domenico Prattichizzo
1.1 Introduction 3
1.1.1 Motivation and Related Works 3
1.1.2 Contributions 4
1.1.3 Organization 5
1.2 Planar Mirrors and Perspective Projection 5
1.2.1 The Virtual Point and the Reflection Transformation 5
1.2.2 The Virtual Camera and the Projection Equivalence 7
1.2.3 Reflective Epipolar Geometry 7
1.3 Single-View and Multiple-view Geometry for PCS Sensors 9
1.3.1 Single-View Geometry 9
1.3.2 Multiple-view Geometry 10
1.4 Mirror Calibration and Image-Based Camera Localization 12
1.4.1 Mirror Calibration 12
1.4.2 Image-Based Camera Localization 13
1.5 Simulation and Experimental Results 16
1.5.1 Simulations 16
1.5.2 Experiments 18
1.6 Conclusions and Future Work 20
References 20

2 Empirical Characterization of Convergence Properties for Kernel-Based Visual Servoing 23
John P. Swensen, Vinutha Kallem, Noah J. Cowan
2.1 Featureless Visual Servoing 23
2.2 Kernel-Based Visual Servoing 25
2.3 Empirical Validation 29
2.3.1 Analysis of the Domain of Attraction 30
2.3.2 Experimental Results 32
2.3.3 Final Error in Pixels 36
2.4 Conclusions 37
References 37

3 High-Speed Visual Feedback Control for Grasping and Manipulation 39
Akio Namiki, Taku Senoo, Satoru Mizusawa, Masatoshi Ishikawa
3.1 Introduction 39
3.2 High-Speed Batting 40
3.2.1 Hybrid Trajectory Generator 40
3.2.2 System Configuration 41
3.2.3 Batting Algorithm 43
3.2.4 Experiment 45
3.3 Tool Manipulation by Visual Servoing 48
3.3.1 Passive Joint in the Contact Point 49
3.3.2 Visual Servo Control for Tool Manipulation 49
3.3.3 Experiments 51
3.4 Conclusion 52
References 52

4 Human-Machine Cooperative Manipulation with Vision-Based Motion Constraints 55
Gregory D. Hager
4.1 Introduction 55
4.2 Virtual Fixtures 56
4.2.1 Virtual Fixtures as a Control Law 57
4.2.2 Virtual Fixtures as Geometric Constraints 58
4.2.3 Choosing the Preferred Direction 59
4.3 Vision-Based Virtual Fixtures 61
4.3.1 Controlling the Viewer: Pure Translation 61
4.3.2 Controlling the Viewer: General Case 64
4.3.3 More General Camera Configurations 65
4.4 Examples of Use 66
4.5 Conclusion 68
References 69

5 Luminance: A New Visual Feature for Visual Servoing ... 71
Christophe Collewet, Eric Marchand
5.1 Introduction ... 71
5.2 Luminance as a Visual Feature ... 73
5.3 Visual Servoing Control Law ... 77
5.3.1 Visual Servoing as an Optimization Problem ... 77
5.3.2 Shape of the Cost Function ... 79
5.3.3 Control Law ... 80
5.4 Experimental Results ... 81
5.4.1 Positioning Tasks under Temporal Luminance Constancy ... 81
5.4.2 Positioning Tasks under Complex Illumination ... 86
5.5 Conclusion and Future Works ... 88
References ... 88

6 Visual Servoing for Beating Heart Surgery ... 91
Wael Bachta, Pierre Renaud, Ezio Malis, Koichi Hashimoto, Jacques Gangloff
6.1 Introduction ... 91
6.2 Motion Compensation Approaches ... 93
6.2.1 Heart-Tool Synchronization ... 94
6.2.2 Heart Immobilization ... 95
6.3 Heart Motion Prediction ... 97
6.3.1 Linear Parameter Varying Method ... 97
6.3.2 Amplitude Modulation Method ... 98
6.4 Robust Real-Time Visual Measurement ... 100
6.4.1 Problem Formulation ... 101
6.4.2 Algorithm ... 102
6.4.3 High-Speed Implementation ... 103
6.4.4 Results ... 103
6.5 Control Strategies ... 105
6.5.1 Heart-Tool Synchronization ... 105
6.5.2 Heart Immobilization ... 108
6.6 Perspectives ... 111
References ... 112

Part II: Estimation and Path-Planning

7 A Variational Approach to Trajectory Planning in Visual Servoing ... 117
Youcef Mezouar
7.1 Introduction ... 117
7.2 Preliminaries ... 119
7.2.1 Brief Review of $SO(3)$ ... 119

7.2.2 Camera Model and Two Views Geometry 120
7.2.3 The Unconstrained Problem 121
7.3 The Constrained Problem 123
7.4 Solving the Variational Problem 125
7.5 Image Space Trajectories 126
7.6 Example 127
7.7 Conclusion 130
References 131

8 Estimation of Homography Dynamics on the Special Linear Group 133
Ezio Malis, Tarek Hamel, Robert Mahony, Pascal Morin
8.1 Introduction 133
8.2 Theoretical Background 135
8.3 Nonlinear Observer on $SL(3)$ 136
8.4 Simulations with Ground Truth 144
8.5 Experiments with Real Data 146
8.6 Conclusion 148
References 149

9 Image Measurement Errors in Visual Servoing: Estimating the Induced Positioning Error 151
Graziano Chesi, Yeung Sam Hung, Ho Lam Yung
9.1 Introduction 151
9.2 Preliminaries 152
9.2.1 Notation and Problem Statement 153
9.2.2 Mathematical Formulation of the Problem 153
9.2.3 SMR of Polynomials 155
9.3 Computation of the Bounds 156
9.3.1 Upper Bounds 157
9.3.2 Lower Bounds 160
9.4 Examples 161
9.4.1 Example 1 161
9.4.2 Example 2 164
9.4.3 Example 3 164
9.5 Conclusion 165
References 165

10 Multicriteria Analysis of Visual Servos through Rational Systems, Biquadratic Lyapunov Functions, and LMIs 169
Patrick Danès, Daniel F. Coutinho, Sylvain Durola
10.1 Introduction 169
10.2 The Rational Systems Framework to Visual Servos Multicriteria Analysis and Synthesis 170

10.2.1 State Space Formulation 171
10.2.2 The Rational Systems Framework 172
10.3 Theoretical Foundations of the Method 174
10.3.1 Elements of Lyapunov Theory 174
10.3.2 Matrix Inequalities and Related Important Lemmas 176
10.4 Multicriteria Analysis through Biquadratic Lyapunov Functions 177
10.4.1 Mathematical Background 177
10.4.2 LMI Conditions for Multicriteria Analysis Based on BQLFs 178
10.4.3 LMI Conditions for Multicriteria Analysis Based on PW-BQLFs 181
10.5 Case Study 184
10.6 Conclusion 186
References 187

11 Path-Planning for Visual Servoing: A Review and Issues 189
Moslem Kazemi, Kamal Gupta, Mehran Mehrandezh
11.1 Introduction 189
11.2 Constraints in Visual Servoing 191
11.2.1 Image/Camera Constraints 191
11.2.2 Robot/Physical Constraints 192
11.3 Path-Planning for Visual Servoing 193
11.3.1 Image Space Path-Planning 193
11.3.2 Optimization-Based Path-Planning 196
11.3.3 Potential Field-Based Path-Planning 198
11.3.4 Global Path-Planning 200
11.4 Path-Planning under Uncertainty for Visual Servoing 201
11.5 Conclusions 203
References 204

12 Single Camera Structure and Motion Estimation 209
Ashwin P. Dani, Warren E. Dixon
12.1 Introduction 209
12.2 Euclidean and Image Space Relationships 211
12.3 Perspective Camera Motion Model 214
12.4 Structure and Motion Estimation 215
12.4.1 Estimation with Known Angular and Linear Velocities 216
12.4.2 Estimation with a Known Linear Velocity 219
References 227

13 Visual Servoing and Pose Estimation with Cameras Obeying the Unified Model 231
Omar Tahri, Youcef Mezouar, François Chaumette, Helder Araujo
13.1 Introduction 231
13.2 Camera Model 234
13.3 Mathematical Background 234
13.3.1 Visual Servoing and Pose Estimation 234
13.3.2 Moments from the Projection of Points onto the Surface of Unit Sphere 236
13.4 Features Choice 238
13.4.1 Invariants to Rotational Motion 238
13.4.2 Variation of the Interaction Matrix with respect to the Camera Pose 238
13.4.3 Features Selection 241
13.5 Results 241
13.5.1 Variation of the Interaction Matrix with respect to Depth Distribution 242
13.5.2 Visual Servoing Results 242
13.5.3 Pose Estimation Results 244
13.6 Conclusion 248
References 248

Part III: Control

14 Gradient Projection Methods for Constrained Image-Based Visual Servo 253
Wenzhuo Ma, Seth Hutchinson
14.1 Introduction 253
14.2 Exploiting Redundancy by Projecting onto Null Spaces 254
14.2.1 Task-Decomposition Approach 254
14.2.2 The Gradient Projection Method Applied to Robotic Control 256
14.3 Gradient Projection Method for Avoiding Joint Limits 259
14.3.1 Basic Algorithm Design 259
14.3.2 Simulation Results 265
14.4 Gradient Projection Method for Occlusion Avoidance 266
14.4.1 Basic Algorithm Design 266
14.4.2 Simulation Results 269
14.5 Gradient Projection Method for Keeping Target Features in the Field of View 270
14.5.1 Basic Algorithm Design 270
14.5.2 Simulation Results 271
14.6 Conclusions 273
References 274

15 Image-Based Visual Servo Control Design with Multi-Constraint Satisfaction 275
Sophie Tarbouriech, Philippe Souères
15.1 Introduction 275
15.2 Problem Statement 277
15.2.1 System Description 277
15.2.2 Problem Formulation 280
15.3 Preliminary Results 280
15.4 Control Design Results 282
15.4.1 Theoretical Issues 283
15.4.2 Optimization Issues 287
15.5 Application 287
15.6 Concluding Remarks and Perspectives 292
References 293

16 Points-Based Visual Servoing with Central Cameras 295
Hicham Hadj-Abdelkader, Youcef Mezouar, Philippe Martinet
16.1 Introduction 295
16.2 Modeling 296
16.2.1 Generic Projection Model 297
16.2.2 Scaled Euclidean Reconstruction 298
16.3 Visual Servoing 300
16.3.1 Task Function and Interaction Matrices 300
16.3.2 Interaction Matrix for 2D Point 301
16.3.3 Decoupled Visual Servoing 302
16.4 Results 305
16.5 Conclusion 311
References 312

17 Sensor-Based Trajectory Deformation: Application to Reactive Navigation of Nonholonomic Robots 315
Florent Lamiraux, Olivier Lefebvre
17.1 Introduction 315
17.2 Nonholonomic Trajectory Deformation as a Dynamic Control System 317
17.2.1 Admissible Trajectories 317
17.2.2 Admissible Trajectory Deformation 317
17.2.3 Potential Field and Inner Product 319
17.3 Nonholonomic Trajectory Deformation Algorithm 320
17.3.1 Finite-Dimensional Subspace of Input Perturbations 321
17.3.2 Boundary Conditions 322
17.3.3 Direction of Deformation That Makes Trajectory Scalar Value Decrease 322
17.3.4 Nonholonomic Constraint Deviation 324

17.4 Application to Mobile Robot Hilare 2 Towing a Trailer 327
17.4.1 Experimental Results 329
17.5 Extension to Docking 329
17.5.1 Docking Task 329
17.5.2 Computation of the Docking Configuration 330
17.6 Experimental Results 331
17.6.1 Bad Localization 332
17.6.2 The Unloading Platform Has Been Moved 333
References 333

18 Unicycle-Like Robots with Eye-in-Hand Monocular Cameras: From PBVS towards IBVS 335
Daniele Fontanelli, Paolo Salaris, Felipe A.W. Belo, Antonio Bicchi
18.1 Introduction 336
18.2 Problem Definition 337
18.2.1 Position-Based Visual Servoing 339
18.2.2 Image-Based Visual Servoing 340
18.2.3 Optimal Trajectories 341
18.3 PBVS in the Large 341
18.3.1 Robot Localization 341
18.3.2 Visual Servoing with Omnidirectional Sight 343
18.3.3 Visual Servoing with FOV Constraint 344
18.3.4 Visual Servoing in the Large 344
18.3.5 Experimental Results 345
18.4 Optimal Trajectories 350
18.4.1 IBVS and Optimal Paths 352
18.4.2 Experimental Results 356
18.5 Conclusions 357
References 358

19 Unmanned Helicopter Control via Visual Servoing with Occlusion Handling 361
Yasushi Iwatani, Kei Watanabe, Koichi Hashimoto
19.1 Introduction 361
19.2 Experimental Setup 362
19.3 Coordinate Frames 364
19.4 Image Jacobian Matrices 365
19.5 Image Feature Estimation and Selection 366
19.5.1 Image Feature Estimation 366
19.5.2 Image Feature Selection 367
19.6 Visual Servo Control with Occlusion Handling 367
19.7 Controller Design 369
19.8 Experimental Result 370
19.9 Conclusions 373
References 373

20 Visual Servoing via Nonlinear Predictive Control 375
Guillaume Allibert, Estelle Courtial, François Chaumette
20.1 Introduction 375
20.2 Predictive Control for Constrained IBVS 377
20.2.1 Visual Predictive Control 377
20.2.2 Internal Model Control Structure 378
20.2.3 Mathematical Formulation 378
20.3 Model of Image Prediction 380
20.3.1 Nonlinear Global Model 381
20.3.2 Local Model Based on the Interaction Matrix 382
20.4 Simulation Results 383
20.4.1 Case 1: Pure Rotation around the Optical Axis 384
20.4.2 Case 2: Large Displacement 389
20.5 Conclusions 392
References 392

Editors' Biographies 395

Index 397

List of Contributors

Guillaume Allibert
Institut PRISME, Polytech'Orleans, 8 rue Leonard de Vinci, 45072 Orleans, France,
guillaume.allibert@univ-orleans.fr

Helder Araujo
Institute for Systems and Robotics, Polo II 3030-290 Coimbra, Portugal,
helder@isr.uc.pt

Wael Bachta
LSIIT, University of Strasbourg, Bd Sebastien Brant, 67412 Illkirch, France,
bachta@lsiit.u-strasbg.fr

Felipe A.W. Belo
Department of Electrical Systems and Automation, University of Pisa, Via Diotisalvi 2, 56100 Pisa, Italy,
felipebelo@gmail.com

Antonio Bicchi
Department of Electrical Systems and Automation, University of Pisa, Via Diotisalvi 2, 56100 Pisa, Italy
bicchi@ing.unipi.it

François Chaumette
INRIA, Campus de Beaulieu, 35042 Rennes, France
francois.chaumette@irisa.fr

Graziano Chesi
Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong
chesi@eee.hku.hk

Christophe Collewet
INRIA Rennes - Bretagne Atlantique, IRISA, Lagadic team, France
christophe.collewet@inria.fr

Estelle Courtial
Institut PRISME, Polytech'Orleans, 8 rue Leonard de Vinci, 45072 Orleans, France
estelle.courtial@univ-orleans.fr

Daniel F. Coutinho
Group of Automation and Control Systems, PUC-RS, Av. Ipiranga 6681, Porto Alegre, RS 90619, Brazil
dcoutinho@pucrs.br

Noah J. Cowan
Department of Mechanical Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
ncowan@jhu.edu

Patrick Danès
CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France;
and
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
patrick.danes@laas.fr

Ashwin P. Dani
Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32608, USA
ashwin31@ufl.edu

Warren E. Dixon
Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32608, USA
wdixon@ufl.edu

Sylvain Durola
CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France
and
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
sylvain.durola@laas.fr

Daniele Fontanelli
Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 14, 38123 Povo (TN), Italy
fontanelli@disi.unitn.it

Jacques Gangloff
LSIIT, University of Strasbourg, Bd Sebastien Brant, 67412 Illkirch, France
jacques.gangloff@lsiit.u-strasbg.fr

Kamal Gupta
Simon Fraser University, Burnaby, BC, V5A1S6, Canada
kamal@cs.sfu.ca

Hicham Hadj-Abdelkader
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France
hadj@lasmea.univ-bpclermont.fr

Gregory D. Hager
Department of Computer Science, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
hager@cs.jhu.edu

Tarek Hamel
I3S-CNRS, 2000 Route des Lucioles, 06903 Sophia Antipolis, France
thamel@i3s.unice.fr

Koichi Hashimoto
Tohoku University, Aoba-ku Aramaki
Aza Aoba 6-6-01, Sendai, Japan
koichi@ic.is.tohoku.ac.jp

Yeung Sam Hung
Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong
yshung@eee.hku.hk

Seth Hutchinson
Department of Electrical and Computer Engineering, University of Illinois, 405 North Matthews Avenue, Urbana, IL

61801, USA
seth@illinois.edu

Masatoshi Ishikawa
Department of Information Physics and Computing, University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan
masatoshi_ishikawa@ipc.i.u-tokyo.ac.jp

Yasushi Iwatani
Tohoku University, Aoba-ku Aramaki Aza Aoba 6-6-01, Sendai, Japan
iwatani@ic.is.tohoku.ac.jp

Vinutha Kallem
GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104, USA
vkallem@seas.upenn.edu

Moslem Kazemi
Simon Fraser University, Burnaby, BC, V5A1S6, Canada
moslemk@cs.sfu.ca

Florent Lamiraux
LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France
florent@laas.fr

Olivier Lefebvre
LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France
olefebr@laas.fr

Wenzhou Ma
Department of Electrical and Computer Engineering, University of Illinois, 405 North Matthews Avenue, Urbana, IL 61801, USA

Robert Mahony
Department of Engineering, Australian National University, North Road, Acton ACT 0200, Australia
Robert.Mahony@anu.edu.au

Ezio Malis
INRIA, 2004 Route des Lucioles, 06902 Sophia Antipolis, France
ezio.malis@sophia.inria.fr

Eric Marchand
INRIA Rennes - Bretagne Atlantique, IRISA, Lagadic team, France
eric.marchand@inria.fr

Gian L. Mariottini
Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455, USA
gianluca@cs.umn.edu

Philippe Martinet
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France
martinet@lasmea.univ-bpclermont.fr

Mehran Mehrandezh
Faculty of Engineering and Applied Science, University of Regina, Regina, Saskatchewan S4S0A2
mehran.mehrandezh@uregina.ca

Youcef Mezouar
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France
mezouar@univ-bpclermont.fr

Satoru Mizusawa
Department of Information Physics and Computing, University of Tokyo,

7-3-1 Hongo, Bunkyo, Tokyo, Japan
satoru_mizusawa@ipc.i.u-tokyo.ac.jp

Fabio Morbidi
Department of Information Engineering, University of Siena, Via Roma 56, 53100 Siena, Italy
morbidi@dii.unisi.it

Pascal Morin
INRIA, 2004 Route des Lucioles, 06902 Sophia Antipolis, France
pascal.morin@sophia.inria.fr

Akio Namiki
Department of Mechanical Engineering, Chiba University, 1-33 Yayoi-cho, Inage, Chiba, Japan
namiki@faculty.chiba-u.jp

Domenico Prattichizzo
Department of Information Engineering, University of Siena, Via Roma 56, 53100 Siena, Italy
prattichizzo@dii.unisi.it

Pierre Renaud
LSIIT, University of Strasbourg, Bd Sebastien Brant, 67412 Illkirch, France
pierre.renaud@lsiit.u-strasbg.fr

Paolo Salaris
Department of Electrical Systems and Automation, University of Pisa, Via Diotisalvi 2, 56100 Pisa, Italy
salarispaolo@gmail.com

Stefano Scheggi
Department of Information Engineering, University of Siena, Via Roma 56, 53100 Siena, Italy
scheggi@dii.unisi.it

Taku Senoo
Department of Information Physics and Computing, University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo, Japan
taku_senoo@ipc.i.u-tokyo.ac.jp

Philippe Souères
LAAS-CNRS, Université de Toulouse, 7 Avenue du Colonel Roche, 31077 Toulouse cedex 4, France
soueres@laas.fr

John P. Swensen
Department of Mechanical Engineering, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
jpswensen@jhu.edu

Omar Tahri
Institute for Systems and Robotics, Polo II 3030-290 Coimbra, Portugal
omartahri@isr.uc.pt

Sophie Tarbouriech
LAAS-CNRS, Université de Toulouse, 7 Avenue du Colonel Roche, 31077 Toulouse cedex 4, France
tarbour@laas.fr

Kei Watanabe
Tohoku University, Aoba-ku Aramaki Aza Aoba 6-6-01, Sendai, Japan
watanabe@ic.is.tohoku.ac.jp

Ho Lam Yung
Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong
h0440266@eee.hku.hk

Abbreviations

AOI	Area of interest
ARIMAX	Autoregressive and integrated moving average with exogenous input
BQLF	Biquadratic Lyapunov function
CABG	Coronary artery bypass grafting
CCD	Charge-coupled device
CISST	Center for computer integrated surgical systems
CPV	Column-parallel high speed vision system
DA	Digital analog
DAR	Differential algebraic representation
DDR	Differential drive robot
DOF	Degrees of freedom
ECG	Electrocardiogram
EKF	Extended Kalman filter
ERSP	Evolution robotics software platform
ESM	Efficient second order minimization
FLC	Fourier linear combiner
FOV	Field of view
GN	Gauss–Newton
GPC	Generalized predictive controller
GPS	Global positioning system
IBO	Identifier-based observer
IBVS	Image-based visual servoing
IMC	Internal model control
IMU	Inertial measurement unit
JHU	Johns Hopkins University
MfS	Motion from structure
MIS	Minimally invasive surgery
MLM	Modified Levenberg–Marquardt
MRI	Magnetic resonance imaging
KBVS	Kernel-based visual servoing

KLT	Kanade–Lucas–Tomasi
LED	Light emitting diode
LMI	Linear matrix inequality
OECD	Organization for economic cooperation and development
OFCE	Optical flow constraint equation
PBVS	Position-based visual servoing
PC	Personal computer
PCI	Percutaneous coronary intervention
PCM	Perceptual control manifold
PCS	Planar catadioptric stereo
PD	Proportional derivative
PID	Proportional integral derivative
PW-BQLF	Piecewise-biquadratic Lyapunov function
RLS	Recursive least-squares
SaM	Structure and motion
SfM	Structure from motion
SHR	Steady hand robot
SDP	Semidefinite programming
SIFT	Scale-invariant feature transform
SLAM	Simultaneous localization and mapping
SMR	Square matricial representation
SOS	Sum of squares of polynomials
SQP	Sequential quadratic program
TECAB	Totally endoscopic coronary artery bypass
UUB	Uniformly ultimately bounded
VPC	Visual predictive control
VSLAM	Visual simultaneous localization and mapping
VVS	Virtual visual servoing

Chapter 1
Catadioptric Stereo with Planar Mirrors: Multiple-view Geometry and Camera Localization

Gian Luca Mariottini, Stefano Scheggi, Fabio Morbidi, and Domenico Prattichizzo

Abstract. Planar catadioptric stereo (PCS) vision sensors consist of a pinhole camera and two or more planar mirrors. PCS systems have recently received an increasing attention in computer vision and have a promising applicability in robotics, since the use of mirrors allows to obtain a stereo view without the need of exact multi-camera synchronization and stereo calibration. The chapter presents a rigorous analytical treatment of the imaging geometry of PCS sensors and introduce new multiple-view properties that are instrumental in addressing the camera localization problem. Original results on mirror calibration are also provided. Extensive simulation and real-data experiments conducted with an eye-in-hand robot illustrate the theory and show the effectiveness of the proposed designs.

1.1 Introduction

In this section we present the motivation for this work, review the related literature, list the original contributions and briefly illustrate the organization of the chapter.

1.1.1 Motivation and Related Works

Stereoscopic vision has been playing a key role in many fields, such as, *e.g.*, 3D imaging for entertainment and mapping, medicine, industrial robotics, augmented reality [13, 11]. Standard stereo sensors are composed of two coupled pinhole cameras and are generally expensive and difficult to calibrate. In addition, the limited field of view considerably limits their application range. To alleviate this problem,

Gian Luca Mariottini
Department of Computer Science and Engineering, University of Minnesota,
200 Union Street SE, Minneapolis, MN 55455, USA
e-mail: gianluca@cs.umn.edu

Stefano Scheggi, Fabio Morbidi, and Domenico Prattichizzo
Department of Information Engineering, University of Siena,
Via Roma 56, 53100 Siena, Italy
e-mail: {scheggi,morbidi,prattichizzo}@dii.unisi.it

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 3–21.
springerlink.com

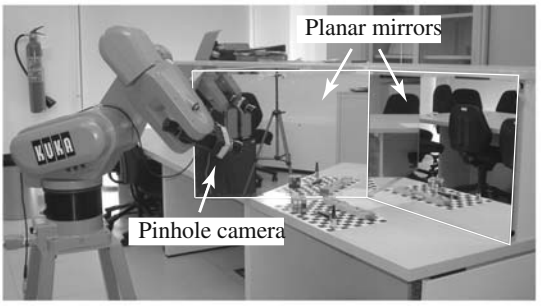


Fig. 1.1 PCS experimental setup composed of a pinhole camera mounted on the end-effector of a manipulator and two planar mirrors

several works in the robotics and computer vision literature have recently proposed the use of *catadioptric* vision sensors [18, 4, 21, 15], which, combining both refracting (lens) and reflecting (mirrors) elements, naturally enlarge the field of view of classic pinhole cameras.

By using mirror reflections of a scene, stereo images can be captured with a single camera (catadioptric stereo). This chapter focuses on a special class of single camera stereo systems called PCS sensors, consisting of a pinhole camera and two, or more, planar mirrors [8, 19]. Single camera stereo has several advantages over traditional two-camera stereo. First of all, only a single set of internal camera calibration parameters needs to be determined. Second, no additional software or hardware is needed for exact camera synchronization [10]. The necessity of possibly large and unwieldy planar mirrors makes PCS systems not suited for mobile robotics. However, these sensors find a natural application in industrial robotics; in fact multiple planar mirrors disposed in the workspace can be used to improve the pose accuracy of an eye-in-hand robotic arm performing high precision assembling and close manipulation tasks.

Several PCS sensors have been designed and implemented in the last few years (see, *e.g.*, [20, 3]) and preliminary results on the geometry, calibration and 3D scene reconstruction from reflective symmetry have been presented in [5, 6, 17]. The set of constraints to be satisfied in order to obtain rectified stereo images from a catadioptric system has been studied in [7]. An affine approximation for epipolar-based PCS rectification has been recently proposed in [22].

From the previous overview, we see that although some research has been done in this field, the specific properties of PCS systems have been addressed only sparsely in the literature and a systematic theory on PCS sensors has yet to be established. In particular, the case of a moving camera in a fixed mirrors scenario and the multiple-view geometry associated with it, have not been explored so far.

1.1.2 Contributions

The original contribution of this chapter is threefold:

- we provide a rigorous analytical treatment of the imaging geometry of PCS sensors composed of a pinhole camera and two planar mirrors (see Figure 1.1) and

introduce new algebraic results on the multiple-view geometry for the case of static and moving cameras (multi-view PCS);
- on the grounds of these novel results, we address the *image-based camera localization* problem and present original methods for *mirror calibration* (*i.e.*, computation of the angle between the mirrors); and
- we present extensive simulation and real-data experiments conducted with an eye-in-hand robot, in order to illustrate the theory and show the effectiveness of the proposed designs in real scenarios.

A preliminary version of this chapter appeared in [16], compared to which we provide here new theoretical results as well as a more extensive experimental validation.

1.1.3 Organization

The rest of this chapter is organized as follows. Section 1.2 reviews the basic theory related to perspective projection through planar mirrors and introduces the reflective epipolar geometry. Section 1.3 deals with the single and multiple-view geometry for PCS sensors. Section 1.4 addresses a solution to the mirror calibration and image-based camera localization problems. Simulation and real-data experiments are reported in Section 1.5. In Section 1.6, the main contributions of the chapter are summarized and possible avenues of future research are highlighted.

1.2 Planar Mirrors and Perspective Projection

In this section we review the imaging properties of catadioptric systems with a single planar mirror [5]. The basic concepts of *virtual point*, *reflection transformation* and *virtual camera projection equivalence* are introduced. The original notion of *reflective epipolar geometry* is presented at the end of the section.

1.2.1 The Virtual Point and the Reflection Transformation

Let us consider the setup reported in Figure 1.2(a) where a perspective camera $\langle c \rangle$ is in front of a planar mirror Π with normal vector $\mathbf{n}_\pi$. A 3D point $\mathbf{X} \triangleq [x\ y\ z]^T$ is supposed to be in front of the mirror as well ($\widetilde{\mathbf{X}}$ indicates its extension in homogeneous coordinates). For the sake of clearness, hereafter we will refer to the simplified setup in Figure 1.2(a): however, the results of this section are valid for generic camera-mirror arrangements. Note that the perspective image $\mathbf{u}$ (pixels) of $\mathbf{X}$ after its reflection by the planar mirror Π can be calculated as the direct projection on $\langle c \rangle$ of the so-called *virtual point* $\mathbf{X}^{[\pi]}$.

Proposition 1.1 (Perspective projection). *Let us consider the setup of Figure 1.2(a) in which a planar mirror with normal vector* $\mathbf{n}_\pi$ *is distant* d_π *from the camera* $\langle c \rangle$. *Then the perspective projection* $\widetilde{\mathbf{u}} \triangleq [u\ v\ 1]^T$ *(pixels) of a generic 3D point* $\mathbf{X}$ *that is mirrored by* Π *is given by*

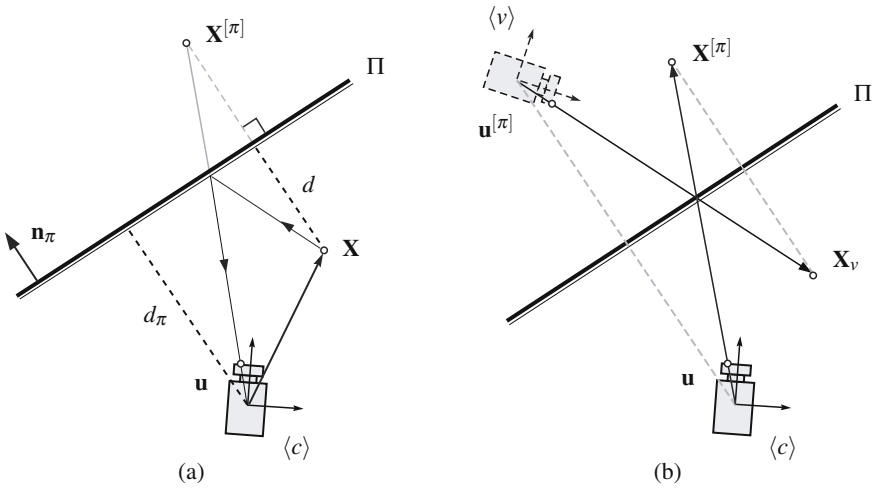


Fig. 1.2 (a) Reflection through the planar mirror Π and the virtual point $\mathbf{X}^{[\pi]}$; and (b) the virtual camera $\langle v \rangle$

$$\lambda\,\widetilde{\mathbf{u}} = \mathbf{K}\,[\mathbf{I}\ \ \mathbf{0}]\,\mathbf{D}^{[\pi]}\,\widetilde{\mathbf{X}} \ \ \textit{with}\ \ \lambda \in \mathbb{R}^{+} \tag{1.1}$$

where $\mathbf{K}$ *is the calibration matrix of the camera and* $\mathbf{D}^{[\pi]}$ *is the* reflection transformation *about the mirror* $\mathbf{n}_\pi$*, defined as*

$$\mathbf{D}^{[\pi]} = \begin{bmatrix} \mathbf{S}^{[\pi]} & 2\,d_\pi\,\mathbf{n}_\pi \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{1.2}$$

where $\mathbf{S}^{[\pi]} = \mathbf{I} - 2\,\mathbf{n}_\pi\mathbf{n}_\pi^T$.

Proof. The perspective projection $\widetilde{\mathbf{u}}$ of the virtual point $\widetilde{\mathbf{X}}^{[\pi]}$ is given by

$$\lambda\,\widetilde{\mathbf{u}} = \mathbf{K}\,[\mathbf{I}\ \ \mathbf{0}]\,\widetilde{\mathbf{X}}^{[\pi]}. \tag{1.3}$$

From geometrical considerations in Figure 1.2(a), we see that

$$\mathbf{X}^{[\pi]} = \mathbf{X} + 2\,d\,\mathbf{n}_\pi. \tag{1.4}$$

Since $d = d_\pi - \mathbf{X}^T\mathbf{n}_\pi$, then (1.4) can be rewritten as

$$\mathbf{X}^{[\pi]} = (\mathbf{I} - 2\,\mathbf{n}_\pi\mathbf{n}_\pi^T)\mathbf{X} + 2\,d_\pi\,\mathbf{n}_\pi. \tag{1.5}$$

By introducing the matrix $\mathbf{D}^{[\pi]}$ defined in (1.2), (1.5) becomes

$$\widetilde{\mathbf{X}}^{[\pi]} = \mathbf{D}^{[\pi]}\,\widetilde{\mathbf{X}}. \tag{1.6}$$

Substituting (1.6) in (1.3), we obtain Equation 1.1. □

Note that $\mathbf{S}^{[\pi]} \in \mathrm{O}(3)$, $\det(\mathbf{S}^{[\pi]}) = -1$ and that $(\mathbf{D}^{[\pi]})^{-1} = \mathbf{D}^{[\pi]}$.

1.2.2 The Virtual Camera and the Projection Equivalence

Proposition 1.1 of Section 1.2.1 shows how the 3D point $\mathbf{X}$ is mirrored by Π at $\mathbf{u}$ onto the image plane of the camera $\langle c \rangle$. Analogously to the concept of virtual point, we can introduce the geometrically intuitive notion of *virtual camera* $\langle v \rangle$ (dashed in Figure 1.2(b)), whose reference frame is simply reflected with respect to $\langle c \rangle$. The proof of the next proposition follows directly from the observation that

$$\mathbf{X}_v = \mathbf{X}^{[\pi]} \tag{1.7}$$

where $\mathbf{X}_v$ is the point $\mathbf{X}$ in $\langle v \rangle$.

Proposition 1.2 (Projection equivalence). *Let* $\mathbf{u}$ *be the perspective projection in* $\langle c \rangle$ *of a 3D point* $\mathbf{X}$ *after its reflection by* Π*. Then,*

$$\mathbf{u} = \mathbf{u}^{[\pi]}, \tag{1.8}$$

being $\mathbf{u}^{[\pi]}$ *the perspective projection of* $\mathbf{X}$ *onto the image plane of the virtual camera* $\langle v \rangle$*.*

Proposition 1.2 states that the perspective projection $\mathbf{u}$ of $\mathbf{X}^{[\pi]}$ coincides with the perspective projection $\mathbf{u}^{[\pi]}$ of $\mathbf{X}_v$. In other words, the camera projections of the reflected points correspond to the virtual camera projections of the real points.

1.2.3 Reflective Epipolar Geometry

In this section we study the imaging geometry relating cameras $\langle c \rangle$ and $\langle v \rangle$. Note that this is different from [5, section 3.1], where the epipolar geometry between the virtual cameras has been investigated.

Proposition 1.3 (Reflective epipolar constraint). *Let us consider the setup in Figure 1.3 and let* d_π *and* $\mathbf{n}_\pi$ *be the distance and the normal of the mirror* Π *measured from* $\langle c \rangle$*, respectively. Let* $\widetilde{\mathbf{u}}_r$*,* $\widetilde{\mathbf{u}}_\pi \in \mathbb{R}^3$ *be the homogeneous representation of the projection of a 3D point in the image plane of the views* $\langle c \rangle$ *and* $\langle v \rangle$*, respectively, and let the camera calibration matrix* $\mathbf{K}$ *be the identity. Then, the* reflective epipolar constraint *is given by*

$$\widetilde{\mathbf{u}}_\pi^T \, \mathbf{E}^{[\pi]} \widetilde{\mathbf{u}}_r = 0$$

where

$$\mathbf{E}^{[\pi]} = 2\, d_\pi \, [\mathbf{n}_\pi]_\times$$

is the reflective essential matrix*, being* $[\mathbf{n}_\pi]_\times$ *the skew-symmetric matrix associated with the vector* $\mathbf{n}_\pi$*.*

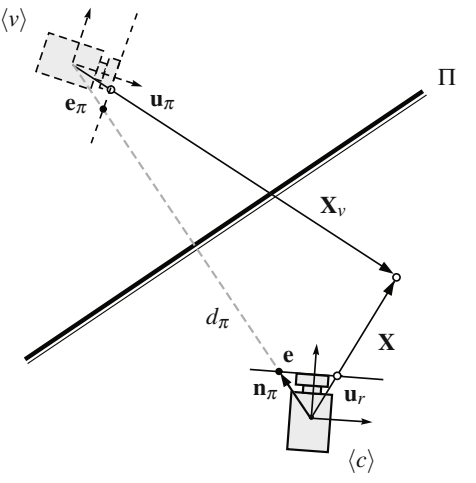


Fig. 1.3 Reflective epipolar geometry

Proof. Let $\mathbf{X}$ and $\mathbf{X}_v$ be the 3D coordinates of a point in the camera frames $\langle c \rangle$ and $\langle v \rangle$, (see Figure 1.3). $\mathbf{X}$ and $\mathbf{X}_v$ are related by the rigid-body transformation

$$\mathbf{X}_v = \mathbf{S}^{[\pi]}\,\mathbf{X} + 2\,d_\pi \mathbf{n}_\pi \tag{1.9}$$

readily derived by substituting (1.7) into (1.5). Since by hypothesis the calibration matrix $\mathbf{K}$ is the identity, then $\mathbf{X}_v = \lambda_\pi \widetilde{\mathbf{u}}_\pi$, $\mathbf{X} = \lambda_r \widetilde{\mathbf{u}}_r$ and (1.9) can be rewritten as

$$\lambda_\pi \widetilde{\mathbf{u}}_\pi = \lambda_r \mathbf{S}^{[\pi]}\,\widetilde{\mathbf{u}}_r + 2\,d_\pi \mathbf{n}_\pi \tag{1.10}$$

where $\lambda_\pi, \lambda_r \in \mathbb{R}^+$ are unknown depths. Simple matrix manipulations on (1.10) lead directly to the epipolar constraint

$$\widetilde{\mathbf{u}}_\pi^T \left(2\,d_\pi\,[\mathbf{n}_\pi]_\times \mathbf{S}^{[\pi]}\right) \widetilde{\mathbf{u}}_r = 0.$$

By definition, $\mathbf{E}^{[\pi]} \triangleq 2\,d_\pi [\mathbf{n}_\pi]_\times \mathbf{S}^{[\pi]} = 2\,d_\pi [\mathbf{n}_\pi]_\times (\mathbf{I} - 2\,\mathbf{n}_\pi \mathbf{n}_\pi^T) = 2\,d_\pi [\mathbf{n}_\pi]_\times$. □

Note that the vector $\mathbf{n}_\pi$ can be readily recovered (up to a scale factor), from the right null-space of $\mathbf{E}^{[\pi]}$. If the camera calibration matrix

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where f_x, f_y (pixels) denote the focal length of the camera along the x and y directions, s is the skew factor and (u_0, v_0) (pixels) are the principal point coordinates of the charge-coupled device (CCD), we can introduce the *reflective fundamental matrix*

$$\mathbf{F}^{[\pi]} \triangleq \mathbf{K}^{-T}\,\mathbf{E}^{[\pi]}\,\mathbf{K}^{-1}. \tag{1.11}$$

Since $\mathbf{F}^{[\pi]}$ is skew-symmetric (in fact we have, $\mathbf{F}^{[\pi]} + (\mathbf{F}^{[\pi]})^T = 2d_\pi(\mathbf{K}^{-T}[\mathbf{n}_\pi]_\times\mathbf{K}^{-1} - \mathbf{K}^{-T}[\mathbf{n}_\pi]_\times\mathbf{K}^{-1}) = \mathbf{0}$), the left and right null-space of $\mathbf{F}^{[\pi]}$ are equal. As such, the epipoles are equal:

$$\widetilde{\mathbf{e}} = \widetilde{\mathbf{e}}_\pi.$$

Note that since $\mathbf{F}^{[\pi]}$ has only 2 degrees of freedom (that correspond to the position of the epipole $\widetilde{\mathbf{e}}$), at least 2 pairs of corresponding points are necessary to determine $\mathbf{F}^{[\pi]}$. Since $\widetilde{\mathbf{e}} = \widetilde{\mathbf{e}}_\pi$ the epipolar lines in the real and virtual view coincide and the epipole can be regarded as a *vanishing point*, being the intersection of 3D parallel lines (the lines joining the real and virtual points) projected onto the image plane.

Remark 1.1. Note that the imaging geometry relating cameras $\langle c \rangle$ and $\langle v \rangle$ corresponds (in terms of essential matrix) to that existing between two cameras undergoing a pure translational motion.

1.3 Single-view and Multiple-view Geometry for PCS Sensors

In this section we introduce the single-view and multiple-view geometry for PCS sensors. These results play a relevant role in the subsequent derivations.

1.3.1 Single-view Geometry

In this section we assume that a camera $\langle c \rangle$ observes a set of 3D points reflected by two planar mirrors (see Figure 1.4). In this case two corresponding virtual cameras $\langle v_1 \rangle$ and $\langle v_2 \rangle$ exist and suitable geometries relating $\langle c \rangle$ with both $\langle v_1 \rangle$ and $\langle v_2 \rangle$ can be defined. The generalization to multiple mirrors is straightforward and it will be not discussed herein. Let be given the image points $\mathbf{u}_i^{[1]}$, $\mathbf{u}_i^{[2]}$, $i \in \{1, \ldots, n\}$ in $\langle c \rangle$, projections of a set of $n \geq 8$ 3D points $\mathbf{X}_i$ reflected onto the mirrors $\mathbf{n}_1$ and $\mathbf{n}_2$, respectively. Note that while the subscript i is the point index, the superscript inside the brackets will always refer to the mirror number through which that vector is reflected (for the sake of simplicity, we will henceforth neglect the subscript π in the mirrors parameters). Given the two-mirror setup reported in Figure 1.4, let $\mathbf{D}^{[1]}$ and $\mathbf{D}^{[2]}$ be the corresponding reflection transformations. The following expression holds true:

$$\mathbf{D}^{[2]}\mathbf{D}^{[1]} = \begin{bmatrix} \mathbf{R}_D & \mathbf{t}_D \\ \mathbf{0}^T & 1 \end{bmatrix}$$

where

$$\mathbf{R}_D \triangleq \mathbf{I} + 4(\mathbf{n}_1^T\mathbf{n}_2)\,\mathbf{n}_1\mathbf{n}_2^T - 2\,\mathbf{n}_1\mathbf{n}_1^T - 2\,\mathbf{n}_2\mathbf{n}_2^T,$$
$$\mathbf{t}_D \triangleq 2d_1\mathbf{n}_1 - 2(d_1(\mathbf{n}_1^T\mathbf{n}_2) + d_2)\mathbf{n}_2.$$

Note that $(\mathbf{R}_D, \mathbf{t}_D)$ only depends on the mirror setup (*i.e.*, $\mathbf{n}_1$, $\mathbf{n}_2$ and d_1, d_2), and not on the observed scene. On the other hand, let $\mathbf{H}_{v_1}^{v_2}$ be the homogeneous transformation matrix representing the rigid body motion between the frames $\langle v_1 \rangle$ and $\langle v_2 \rangle$. It is easily found that (see Figure 1.4)

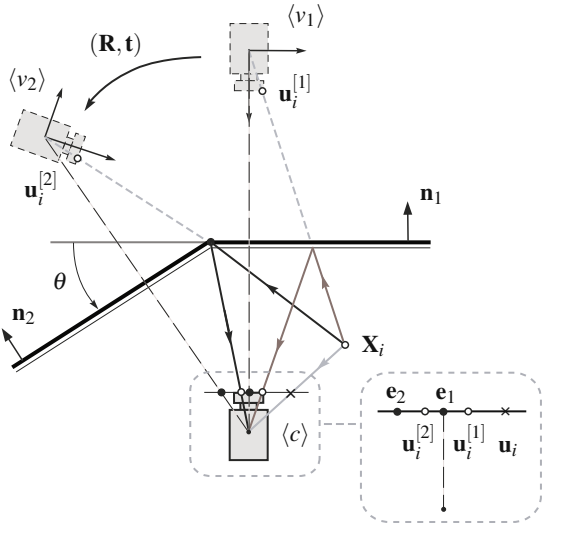


Fig. 1.4 The camera $\langle c \rangle$ and the virtual cameras $\langle v_1 \rangle$ and $\langle v_2 \rangle$

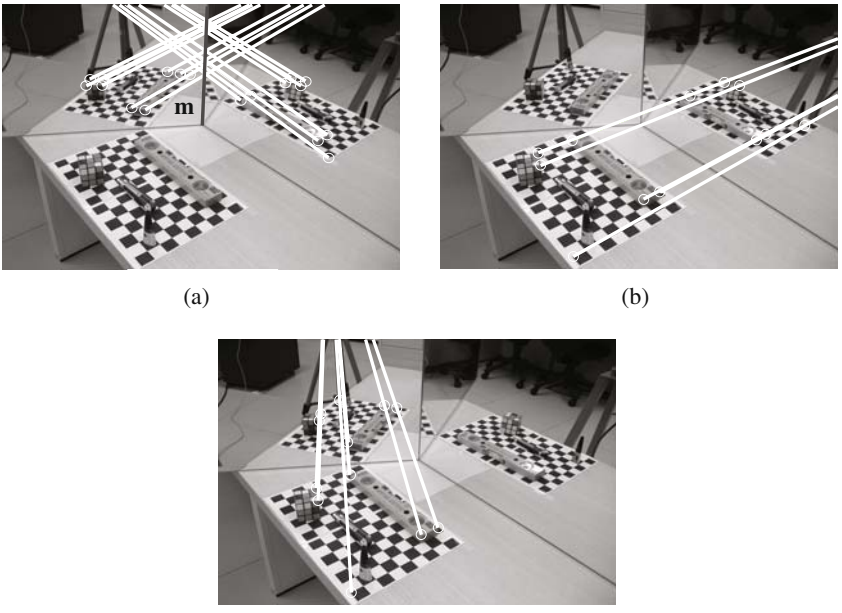
$$\mathbf{H}_{v_1}^{v_2} \triangleq \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} = \mathbf{D}^{[1]} \mathbf{D}^{[2]} \tag{1.12}$$

where $(\mathbf{R}, \mathbf{t})$ is the rigid body motion between $\langle v_1 \rangle$ and $\langle v_2 \rangle$. Owing to Proposition 1.2, the points $\mathbf{u}_i^{[1]}$ and $\mathbf{u}_i^{[2]}$ in $\langle c \rangle$ are corresponding in both the virtual cameras $\langle v_1 \rangle$ and $\langle v_2 \rangle$, (see Figure 1.4). This implies the existence of the epipolar geometry relating $\langle v_1 \rangle$ and $\langle v_2 \rangle$, *i.e.*, $\widetilde{\mathbf{u}}_i^{[2]\,T} \mathbf{F}\, \widetilde{\mathbf{u}}_i^{[1]} = 0$. The fundamental matrix $\mathbf{F}$ can be estimated from a set of (at least) 8 image points and the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$ are obtained as the right and left null-spaces of $\mathbf{F}$ [9]. Moreover, given the camera calibration matrix $\mathbf{K}$, from $\mathbf{F}$ we can compute the essential matrix $\mathbf{E} \triangleq [\mathbf{t}]_\times \mathbf{R}$. Once $\mathbf{E}$ is known, a decomposition [9] can be carried out to finally compute the matrix $\mathbf{R}$ and the vector $\mathbf{t}$ (up to a scale factor).

Figure 1.5 shows the epipolar lines (white) relative to pairs of corresponding points in the real and virtual views, on a sample image. Figure 1.5(a) reports the epipolar lines between the virtual cameras $\langle v_1 \rangle$ and $\langle v_2 \rangle$: the baseline lies far above the edge of the image. As shown in [6], all corresponding epipolar lines intersect at the image projection $\mathbf{m}$ of the mirrors screw axis (*i.e.*, the 3D line of intersection between the mirrors). Figures 1.5(b–c) show the epipolar lines between $\langle v_1 \rangle$ and $\langle c \rangle$, and between $\langle v_2 \rangle$ and $\langle c \rangle$, respectively.

1.3.2 Multiple-view Geometry

In this section we address the case of a moving camera that observes a set of 3D points $\mathbf{X}_i$ reflected by two mirrors, from two views $\langle c_1 \rangle$ and $\langle c_2 \rangle$ (see Figure 1.6). Let $\mathbf{H}_R \triangleq \mathbf{H}_{c_1}^{c_2}$ be the homogeneous transformation matrix relating $\langle c_1 \rangle$ and $\langle c_2 \rangle$, and $\mathbf{H}_{v_1^{[2]}}^{v_2^{[2]}}$, $\mathbf{H}_{v_2^{[1]}}^{v_2^{[2]}}$, $\mathbf{H}_{v_1^{[1]}}^{v_2^{[1]}}$, $\mathbf{H}_{v_1^{[1]}}^{v_1^{[2]}}$ (with a slight abuse of notation since differently



(a) (b)

(c)

Fig. 1.5 Epipolar lines (white) on a sample image: (a) between $\langle v_1 \rangle$ and $\langle v_2 \rangle$; (b) between $\langle v_1 \rangle$ and $\langle c \rangle$; and (c) between $\langle v_2 \rangle$ and $\langle c \rangle$

from Section 1.3.1 the subscript of v refers herein to the camera number), the four homogeneous matrices encoding the relative pose between the virtual views. Finally, let $\mathbf{D}_{c_1}^{[1]}$ and $\mathbf{D}_{c_1}^{[2]}$ denote the reflection transformations about the two mirrors, written in $\langle c_1 \rangle$. We are now ready to introduce the following proposition, which establishes a link between $\mathbf{H}_R$ and $\mathbf{H}_{v_2^{[1]}}^{v_2^{[2]}}$.

Proposition 1.4 (Multiple cameras mirroring). *The following equation holds true:*

$$\mathbf{H}_{v_2^{[1]}}^{v_2^{[2]}} = \mathbf{H}_R^{-1}\, \mathbf{D}_{c_1}^{[1]} \mathbf{D}_{c_1}^{[2]}\, \mathbf{H}_R \tag{1.13}$$

Proof. From (1.12) we have that

$$\mathbf{D}_{c_1}^{[1]} \mathbf{D}_{c_1}^{[2]} = \mathbf{H}_{v_1^{[1]}}^{v_1^{[2]}}. \tag{1.14}$$

Pre-multiplying and post-multiplying (1.14) by $\mathbf{H}_R^{-1}$ and $\mathbf{H}_R$, respectively, we obtain

$$\mathbf{H}_R^{-1} \mathbf{H}_{v_1^{[1]}}^{v_1^{[2]}} \mathbf{H}_R = \mathbf{H}_R^{-1} \mathbf{D}_{c_1}^{[1]} \mathbf{D}_{c_1}^{[2]} \mathbf{H}_R. \tag{1.15}$$

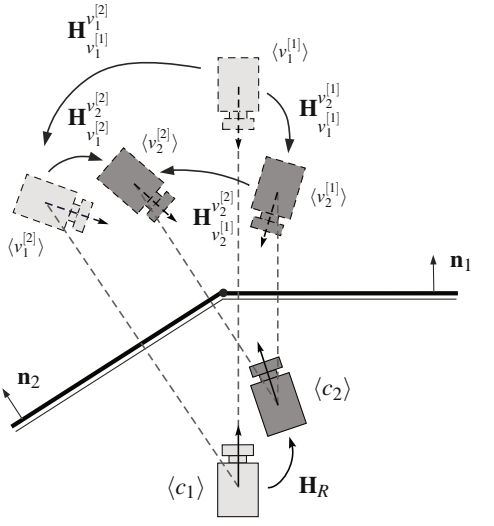


Fig. 1.6 Multiple-view geometry for the cameras $\langle c_1 \rangle$ and $\langle c_2 \rangle$

From an inspection of Figure 1.6, it is easy to verify that the following equalities hold true:

$$\mathbf{H}_R = \mathbf{H}_{v_1^{[1]}}^{v_2^{[1]}} = \mathbf{H}_{v_1^{[2]}}^{v_2^{[2]}} \tag{1.16}$$

i.e., the rigid motion between $\langle c_1 \rangle$ and $\langle c_2 \rangle$ is equal to the rigid motion between $\langle v_1^{[1]} \rangle$ and $\langle v_2^{[1]} \rangle$, and between $\langle v_1^{[2]} \rangle$ and $\langle v_2^{[2]} \rangle$. Using (1.16) and (1.14) into (1.15), we obtain

$$\mathbf{H}_R^{-1}\mathbf{H}_{v_1^{[1]}}^{v_1^{[2]}}\mathbf{H}_R = \mathbf{H}_{v_2^{[2]}}^{v_1^{[2]}}\mathbf{H}_{v_1^{[1]}}^{v_1^{[2]}}\mathbf{H}_{v_1^{[2]}}^{v_2^{[2]}} = \mathbf{H}_{v_2^{[2]}}^{v_1^{[2]}}\mathbf{H}_{v_1^{[1]}}^{v_2^{[1]}}\mathbf{H}_{v_2^{[1]}}^{v_2^{[2]}} = \mathbf{H}_R^{-1}\mathbf{H}_R\mathbf{H}_{v_2^{[1]}}^{v_2^{[2]}} = \mathbf{H}_{v_2^{[1]}}^{v_2^{[2]}},$$

and (1.13) is thus proved. □

Remark 1.2. Note that Equation 1.13 allows one to estimate the rigid motion $\mathbf{H}_R$ also when the epipolar geometry between $\langle c_1 \rangle$ and $\langle c_2 \rangle$ is not well-defined (*small baseline* case). In fact the epipolar geometry between the virtual cameras is always well-defined by construction.

1.4 Mirror Calibration and Image-based Camera Localization

In this section we address the mirror calibration and image-based camera localization problems using the PCS properties presented in Sections 1.2 and 1.3. Proposition 1.5 in the next section will be instrumental for Proposition 1.8 in Section 1.4.2.

1.4.1 Mirror Calibration

Consider the setup in Figure 1.4 and assume that at least $n \geq 2$ points can be directly observed by $\langle c \rangle$ at $\mathbf{u}_i$, $i \in \{1, \ldots, n\}$. If the same set of points is also reflected by

the mirrors $\mathbf{n}_1$ and $\mathbf{n}_2$ at $\mathbf{u}_i^{[1]}$ and $\mathbf{u}_i^{[2]}$ (white dots), then two reflective fundamental matrices $\mathbf{F}^{[1]}$ and $\mathbf{F}^{[2]}$ do exist and can be computed from the corresponding pairs $(\mathbf{u}_i^{[1]}, \mathbf{u}_i)$ and $(\mathbf{u}_i^{[2]}, \mathbf{u}_i)$, respectively (cf. Proposition 1.2). From $\mathbf{F}^{[1]}$ and $\mathbf{F}^{[2]}$, we can then determine the epipoles $\mathbf{e}_1$ and $\mathbf{e}_2$ as their right null-spaces (black dots). Since the direction of each epipole $\mathbf{e}_j$ is always parallel to $\mathbf{n}_j$, $j \in \{1,2\}$, we then obtain the following:

Proposition 1.5 (Mirror calibration with the epipoles). *The angle θ between the mirrors $\mathbf{n}_1$ and $\mathbf{n}_2$ is given by*

$$\theta = \arccos\left(\frac{(\mathbf{K}^{-1}\widetilde{\mathbf{e}}_1)^T(\mathbf{K}^{-1}\widetilde{\mathbf{e}}_2)}{\|\mathbf{K}^{-1}\widetilde{\mathbf{e}}_1\|\,\|\mathbf{K}^{-1}\widetilde{\mathbf{e}}_2\|}\right).$$

In the next proposition the epipoles between the virtual cameras are used to solve the mirror calibration problem (see Figure 1.7(a)). Let $\mathbf{F}^{[21]}$ be the fundamental matrix computed from the corresponding points $(\mathbf{u}_i^{[1]}, \mathbf{u}_i^{[2]})$ and let γ be the angle between the virtual epipoles $\mathbf{e}^{[12]}$ and $\mathbf{e}^{[21]}$. It is easy to verify that

$$\gamma = \arccos\left(\frac{(\mathbf{K}^{-1}\widetilde{\mathbf{e}}^{[12]})^T(\mathbf{K}^{-1}\widetilde{\mathbf{e}}^{[21]})}{\|\mathbf{K}^{-1}\widetilde{\mathbf{e}}^{[12]}\|\,\|\mathbf{K}^{-1}\widetilde{\mathbf{e}}^{[21]}\|}\right).$$

Proposition 1.6 (Mirror calibration with the virtual epipoles). *The angle θ between the mirrors $\mathbf{n}_1$ and $\mathbf{n}_2$ is given by*

$$\theta = \frac{\pi - \gamma}{2}.$$

Remark 1.3. It is worth emphasizing here that our calibration notion is different from that considered in previous works (and notably in [6]). In fact, with "mirror calibration" we mean the estimation of the *angle* between the mirrors, while in [6] the authors mean the estimation the *focal length* of the camera and the orientation of the mirrors *screw axis*.

1.4.2 Image-based Camera Localization

This section deals with the estimation of the rigid motion $(\mathbf{R}_w^c, \mathbf{t}_w^c)$ of the camera $\langle c \rangle$ with respect to a world frame $\langle w \rangle$. In the interest of simplicity, we will assume that the z-axis of $\langle w \rangle$ coincides with the mirrors screw axis and the x-axis lies on mirror $\mathbf{n}_1$.

Proposition 1.7 provides a method to estimate the matrix $\mathbf{R}_w^c$, while Proposition 1.8 presents two methods for computing the projection of $\mathbf{t}_w^c$ on the plane defined by the camera centers. Consider the setup reported in Figures 1.7(a–b). If at least two corresponding points exist between $\langle c \rangle$ and $\langle v_1 \rangle$ and between $\langle c \rangle$ and $\langle v_2 \rangle$, then the fundamental matrices $\mathbf{F}^{[1]}$ and $\mathbf{F}^{[2]}$ exist. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be the epipoles having unitary norm considered in Proposition 1.5, $\mathbf{z}_{w(c)} \triangleq \mathbf{e}_1 \times \mathbf{e}_2$ and $\mathbf{x}_{w(c)} \triangleq \mathbf{e}_1 \times \mathbf{z}_{w(c)}$.

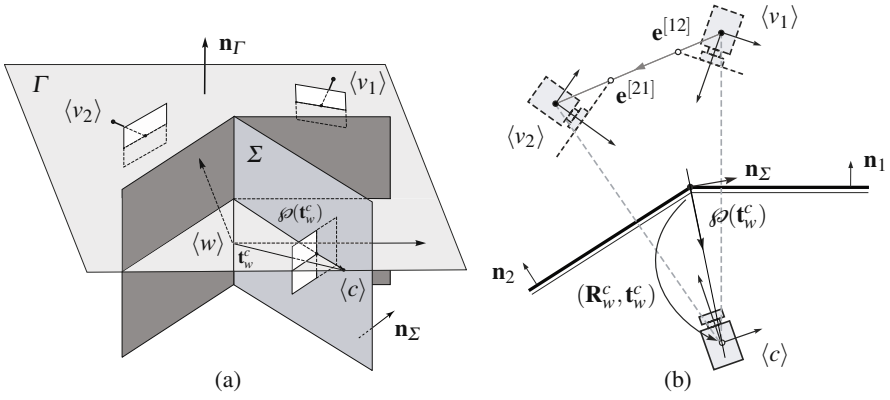


Fig. 1.7 (a) $\wp(\mathbf{t}_w^c)$ is the projection of $\mathbf{t}_w^c$ on the plane Γ; and (b) computation of $\wp(\mathbf{t}_w^c)$ using the epipolar lines and Σ

Proposition 1.7 ($\mathbf{R}_w^c$ estimation). *For every rigid-motion* $(\mathbf{R}_w^c, \mathbf{t}_w^c)$*, the following equation holds true,*

$$\mathbf{R}_w^c = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \left[\mathbf{n}_{\Gamma(c)} \;\; \mathbf{e}_1 \;\; \mathbf{a}_{(c)} \right]^{-1}. \tag{1.17}$$

Proof. Due to the assumption that the x-axis of $\langle w \rangle$ lies on $\mathbf{n}_1$, for every pose of $\langle c \rangle$ the world y-axis expressed in the camera frame $\langle c \rangle$ corresponds to $\mathbf{e}_1$. Consequently, for every pose of $\langle c \rangle$, $\mathbf{e}_1$ and $\mathbf{e}_2$ lie on the same plane Γ defined by the three camera centers having normal vector $\mathbf{n}_{\Gamma_{(w)}} = [0 \;\; 0 \;\; 1]^T$ in $\langle w \rangle$ (see Figure 1.7(a)), then

$$\mathbf{z}_{w(c)} \triangleq \mathbf{e}_1 \times \mathbf{e}_2$$

where $\mathbf{z}_{w(c)}$ is the z-axis of the world reference frame expressed in the camera frame $\langle c \rangle$. The world frame x-axis can be easily obtained as the cross product of $\mathbf{e}_1$ and $\mathbf{z}_{w(c)}$:

$$\mathbf{x}_{w(c)} \triangleq \mathbf{e}_1 \times \mathbf{z}_{w(c)}.$$

Finally, we have that

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} = \mathbf{R}_w^c \left[\mathbf{x}_{w(c)} \;\; \mathbf{e}_1 \;\; \mathbf{z}_{w(c)} \right]$$

from which (1.17) follows. □

Note that, in absence of noise on the image points, Equation 1.17 provides us with the *exact* $\mathbf{R}_w^c$. In the case of noisy data, the estimated $\mathbf{R}_w^c$ will not be, in general,

a rotation matrix. To overcome this problem, the correct rotation matrix should be computed as $\mathbf{U}\mathbf{V}^T$, where matrices $\mathbf{U}$ and $\mathbf{V}$ are obtained from the singular value decomposition of the initial estimate of $\mathbf{R}_w^c$.

Remark 1.4. In Section 1.3.2 we have seen that Equation 1.13 relates the pose of two cameras $\langle c_1 \rangle$ and $\langle c_2 \rangle$ with the pose of their virtual counterparts. An analogous equation, that will be instrumental in proving the second statement in Proposition 1.8, relates the pose of $\langle c \rangle$ and $\langle w \rangle$ with that of the virtual cameras $\langle v_1 \rangle$ and $\langle v_2 \rangle$. In fact, as from Figure 1.6, assuming that the camera frame $\langle c_1 \rangle$ is coincident with $\langle w \rangle$ and $\langle c_2 \rangle$ is coincident with $\langle c \rangle$, we get

$$\mathbf{H}_{v_1}^{v_2} = \mathbf{H}_c^w \mathbf{D}_w^{[1]} \mathbf{D}_w^{[2]} \mathbf{H}_w^c. \tag{1.18}$$

By inverting and then pre-multiplying (1.18) by $\mathbf{H}_w^c$, we obtain the *Sylvester equation* (with unknown $\mathbf{H}_w^c$)

$$\mathbf{H}_w^c \mathbf{H}_{v_2}^{v_1} = \mathbf{D}_w^{[2]} \mathbf{D}_w^{[1]} \mathbf{H}_w^c \tag{1.19}$$

where $\mathbf{D}_w^{[1]}$ and $\mathbf{D}_w^{[2]}$ are the reflection transformations about the two mirrors written in $\langle w \rangle$:

$$\mathbf{D}_w^{[1]} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D}_w^{[2]} = \begin{bmatrix} \mathbf{S}_w^{[2]} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}$$

being $\mathbf{n}_1 = [0 \;\; 1 \;\; 0]^T$ and $\mathbf{n}_2 = [-\sin\theta \;\; \cos\theta \;\; 0]^T$ in $\langle w \rangle$.

Consider the setup in Figure 1.7(a) and let $\wp(\mathbf{t}_w^c) \in \mathbb{R}^2$ be the vector containing the first two components of the projection of $\mathbf{t}_w^c$ on the plane Γ. In the next proposition we present two methods for estimating the direction of $\wp(\mathbf{t}_w^c)$, *i.e.*, $\frac{\wp(\mathbf{t}_w^c)}{\|\wp(\mathbf{t}_w^c)\|}$. The first method uses the fundamental matrix $\mathbf{F}^{[12]}$ between the virtual cameras $\langle v_1 \rangle$, $\langle v_2 \rangle$ and the second one the Equation 1.19. Let Σ be the plane defined by the mirrors screw axis and the center of $\langle c \rangle$, and let $\mathbf{n}_\Sigma$ be the normal to Σ (see Figure 1.7(b)). In the next proposition $\mathbf{R}_z(\gamma)$ denotes a rotation about the z-axis by an angle γ.

Proposition 1.8 (Estimation of $\frac{\wp(\mathbf{t}_w^c)}{\|\wp(\mathbf{t}_w^c)\|}$). *The direction of $\wp(\mathbf{t}_w^c)$ is given by*

$$\frac{\wp(\mathbf{t}_w^c)}{\|\wp(\mathbf{t}_w^c)\|} = \frac{\mathbf{t}_w^{c*}}{\|\mathbf{t}_w^{c*}\|} \tag{1.20}$$

where

$$\mathbf{t}_w^{c*} = [\mathbf{a}(1) \;\; \mathbf{a}(2)]^T \tag{1.21}$$

with $\mathbf{a} = \mathbf{R}_z(-90\,deg)\,\mathbf{n}_\Sigma$, $\mathbf{n}_\Sigma(1) > 0$, *or alternatively,*

$$\mathbf{t}_w^{c*} = \left[\frac{\mathbf{a}(1)(1-\mathbf{n}_2^2(2)) + \mathbf{n}_2(1)\mathbf{n}_2(2)\mathbf{a}(2)}{2\mathbf{n}_2^2(1)} \quad \frac{\mathbf{a}(2)\mathbf{n}_2(1) - \mathbf{a}(1)\mathbf{n}_2(2)}{2\mathbf{n}_2(1)} \right]^T \tag{1.22}$$

with $\mathbf{a} = -\mathbf{R}_w^c \mathbf{t}_{v_2}^{v_1}$, *where* $\mathbf{n}_2$ *is the normal vector to the second mirror, written in* $\langle w \rangle$.

Proof. Let us start by proving the first part of the statement. $\mathbf{t}_w^c$ and $\wp(\mathbf{t}_w^c)$ lie on the plane Σ with normal vector $\mathbf{n}_\Sigma = [\mathbf{n}_\Sigma(1)\ \ \mathbf{n}_\Sigma(2)\ \ 0]^T$ in $\langle w \rangle$. Since the corresponding epipolar lines (computed from the fundamental matrix $\mathbf{F}^{[12]}$) all intersect at a single image line $\mathbf{m}$, projection of the screw axis, we have that

$$\mathbf{n}_\Sigma = \mathbf{R}_w^c \, (\mathbf{K}^T \mathbf{m})$$

where $\mathbf{K}^T\mathbf{m}$ is the normal vector to Σ in $\langle c \rangle$. Since $\mathbf{n}_\Sigma \perp \wp(\mathbf{t}_w^c)$, it is then sufficient to rotate $\mathbf{n}_\Sigma$ of -90 deg around the z-axis in order to obtain $\mathbf{t}_w^{c*}$ in (1.21).

To prove the second part of the statement, consider the Equation 1.19. Let $\mathbf{R}_w^c$ be given (computed, for example, using Proposition 1.7) and $\mathbf{D}_w^{[2]}$, $\mathbf{n}_2$ be estimated using one of the algorithms in Section 1.4.1. Let $\mathbf{H}_{v_2}^{v_1}$ also be given (see Section 1.3.1). From (1.19) we have that $\mathbf{R}_w^c \mathbf{t}_{v_2}^{v_1} + \mathbf{t}_w^c = \mathbf{S}_w^{[2]} \mathbf{S}_w^{[1]} \mathbf{t}_w^c$. Collecting $\mathbf{t}_w^c$ on the left-hand side of the equation, we get $(\mathbf{I} - \mathbf{S}_w^{[2]} \mathbf{S}_w^{[1]}) \mathbf{t}_w^c = -\mathbf{R}_w^c \mathbf{t}_{v_2}^{v_1}$, that is

$$\begin{bmatrix} 2\mathbf{n}_2^2(1) & -2\mathbf{n}_2(1)\mathbf{n}_2(2) & 0 \\ 2\mathbf{n}_2(1)\mathbf{n}_2(2) & 2(1-\mathbf{n}_2^2(2)) & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{t}_w^c = -\mathbf{R}_w^c \mathbf{t}_{v_2}^{v_1}$$

from which, after few manipulations, we obtain (1.22). Note that the normalization of $\mathbf{t}_w^{c*}$ in (1.20), removes the ambiguity due to the up to scale estimation of $\mathbf{t}_{v_2}^{v_1}$. □

1.5 Simulation and Experimental Results

In this section we present the results of the numerical simulations and real-data experiments we performed to elucidate and validate the theory.

1.5.1 Simulations

Simulation experiments have been conducted with the *epipolar geometry toolbox* [14], in order to test the effectiveness of the algorithms presented in the previous sections. The setup is composed of a pinhole camera with calibration matrix

$$\mathbf{K} = \begin{bmatrix} 951.8 & 0 & 640.66 \\ 0 & 951.8 & 605.11 \\ 0 & 0 & 1 \end{bmatrix}$$

and two planar mirrors with normal vectors $\mathbf{n}_1 = [0.6\ \ 0.55\ \ -0.57]^T$ and $\mathbf{n}_2 = [-0.32\ \ 0.87\ \ -0.37]^T$, corresponding to an angular displacement $\theta = 60$ deg. The camera observes directly a total of 20 random points. For the sake of simplicity, we will henceforth suppose that the correspondence matching problem is exact. In order to solve the correspondence problem in practice, one might first use the

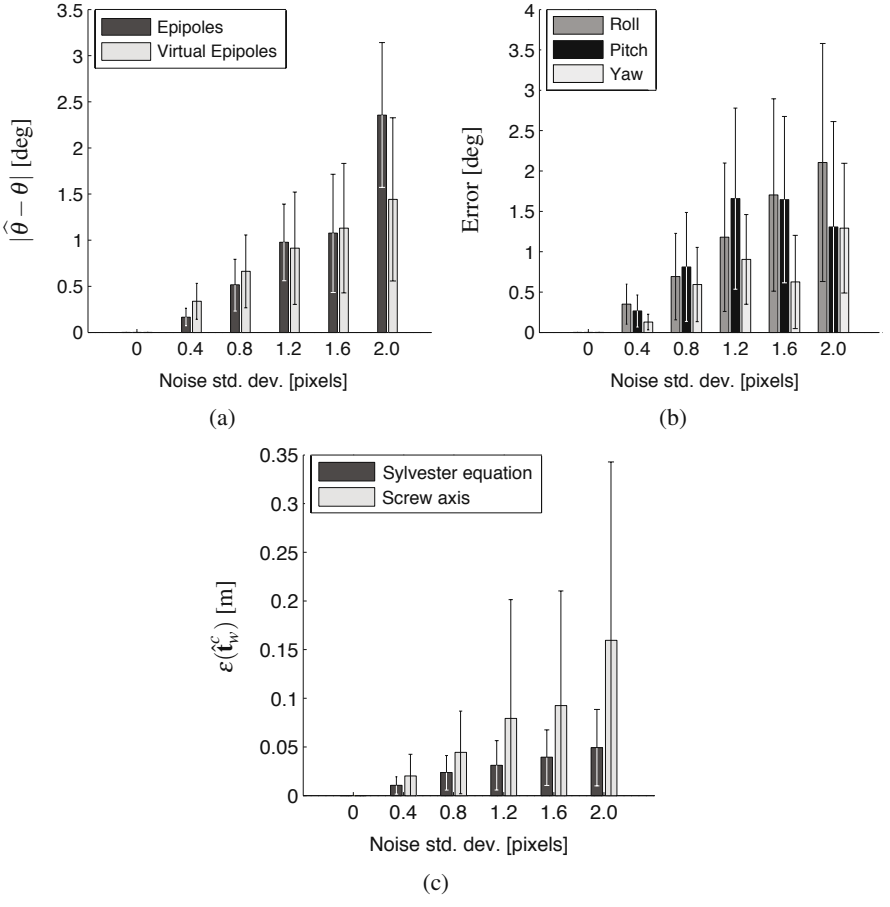


Fig. 1.8 Simulations: (a) mirror angle estimation error $|\widehat{\theta}-\theta|$ for the algorithms in Propositions 1.5 and 1.6; (b) $\mathbf{R}_w^c$ estimation error for the algorithm in Proposition 1.7; and (c) error $\varepsilon(\hat{\mathbf{t}}_w^c)$ for the algorithms in Proposition 1.8

scale-invariant feature transform (SIFT) [12] and, in a second phase, assign the points to each mirror which has been uniquely identified by a colored landmark.

The rigid motion $(\mathbf{R}_w^c, \mathbf{t}_w^c)$ between the camera and world frame is

$$\mathbf{R}_w^c = \mathbf{R}_z(40\ \text{deg})\,\mathbf{R}_y(-20\ \text{deg})\,\mathbf{R}_x(30\ \text{deg}),\ \ \mathbf{t}_w^c = [0.6\ \ -2\ \ -0.2]^T\ \text{m}.$$

Four well-known fundamental matrix estimators [1], *normalized 8-point algorithm*, *iterative Sampson method*, *robust Torr's M-estimator* and *LMedS*, have been compared in the simulation experiments. The M-estimator revealed the best compromise between estimation accuracy, computation load and robustness to uncorrect correspondence matching due to outliers.

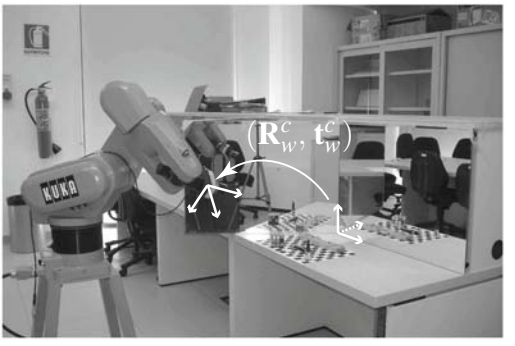


Fig. 1.9 Experimental setup: an eye-in-hand robot observes a structured 3D scene directly and reflected through two planar mirrors

In Figure 1.8, we ran 200 iterations of the proposed algorithms for increasing value of the image noise standard deviation. Figure 1.8(a) reports the mirror angle estimation error $|\widehat{\theta} - \theta|$ for the algorithms in Propositions 1.5 and 1.6. The algorithm that uses the virtual epipoles is the most accurate with a maximum mean error of 1.5 deg. Figure 1.8(b) shows the $\mathbf{R}_w^c$ estimation error computed as the roll-pitch-yaw angle errors (cf. Proposition 1.7). We experienced a good robustness to noise of the proposed method, with a maximum mean error around 2 deg. Figure 1.8(c) finally reports the estimation error

$$\varepsilon(\hat{\mathbf{t}}_w^c) = \left\| \frac{\wp(\hat{\mathbf{t}}_w^c)}{\|\wp(\hat{\mathbf{t}}_w^c)\|} - \frac{\wp(\mathbf{t}_w^c)}{\|\wp(\mathbf{t}_w^c)\|} \right\|$$

for the two algorithms in Proposition 1.8: *screw axis* (1.21) and *Sylvester equation* (1.22). From an inspection of Figure 1.8(c), we observe that the second method is less sensitive to noise since it does not rely on the direct computation of the epipoles. A maximum mean error of 5 cm is achieved.

1.5.2 Experiments

In order to test the robustness of proposed algorithms in real scenarios we performed a series of experiments using the setup shown in Figure 1.9. A Lumenera®[1] LU071C camera mounted on the end-effector of a 6 axes KUKA®[2] KR 3 manipulator, observes a structured scene directly and reflected through two planar mirrors. The mirrored objects lie in a box of size 0.42×0.3 m^2.

We exploited the high positioning accuracy of the KR 3 manipulator in order to have precise ground truth reference measurements of position and orientation of the camera. The calibration matrix $\mathbf{K}$ of the camera is the same as in the simulations[3].

Figure 1.10(a) shows the mirror angle estimation error $|\widehat{\theta} - \theta|$ for $\theta \in \{30 \text{ deg}, 45 \text{ deg}, 60 \text{ deg}\}$. The values reported in Figure 1.10(a) are the average of 5

[1] Lumenera® is a registered trademark of Lumenera Corporation. http://www.lumenera.com
[2] KUKA® is a registered trademark of KUKA AG. http://www.kuka.com
[3] The calibration matrix has been estimated with the *camera calibration toolbox* [2].

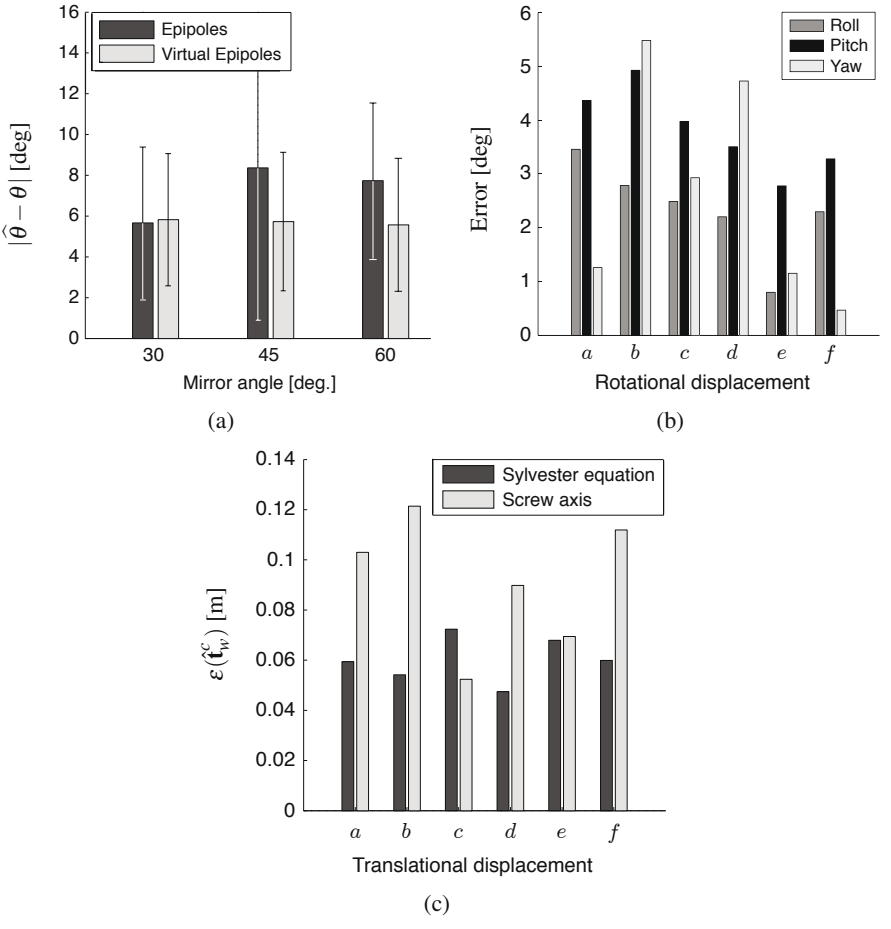


Fig. 1.10 Experiments: (a) mirror angle estimation error $|\widehat{\theta} - \theta|$ for $\theta \in \{30 \text{ deg}, 45 \text{ deg}, 60 \text{ deg}\}$; (b) $\mathbf{R}_w^c$ estimation error for the rotational displacements reported in Table 1.1; and (c) error $\varepsilon(\widehat{\mathbf{t}}_w^c)$ for the translational displacements reported in Table 1.1

experiments conducted for each θ. In confirmation of the simulation results, the algorithm that exploits the virtual cameras epipoles is the most accurate, with a maximum mean error of about 6 deg. Figure 1.10(b) reports the $\mathbf{R}_w^c$ estimation error computed as the roll-pitch-yaw angle errors, for the six rotational displacements reported in Table 1.1, ($\theta = 60$ deg). As already pointed out in Section 1.5.1, the proposed algorithm has a reduced sensitivity to noise: the maximum error is less than 6 deg. Finally, Figure 1.10(c) depicts the estimation error $\varepsilon(\widehat{\mathbf{t}}_w^c)$, for the six translational displacements reported in Table 1.1. As in the simulation experiments, the algorithm based on the Sylvester equation exhibits the best performance, with a maximum error of about 7 cm.

Table 1.1 Rotational and translational displacements considered in Figures 1.10(b–c). The mirror angle $\theta = 60$ deg

	$\mathbf{R}_w^c$			$\mathbf{t}_w^c$
	Roll [deg]	*Pitch* [deg]	*Yaw* [deg]	[*m*]
a	−4.00	−3.03	−37.07	$[\,0.147\ -0.824\ \ 0.377\,]^T$
b	8.16	−3.49	−43.13	$[\,0.157\ -0.824\ \ 0.610\,]^T$
c	13.99	−20.87	−40.57	$[\,0.432\ -0.960\ \ 0.532\,]^T$
d	−6.38	−9.01	−51.60	$[\,0.106\ -0.613\ \ 0.408\,]^T$
e	−3.63	−3.46	−43.73	$[\,0.215\ -0.824\ \ 0.552\,]^T$
f	7.74	−13.55	−35.60	$[\,0.313\ -0.960\ \ 0.535\,]^T$

1.6 Conclusions and Future Work

The chapter presents a systematic study of the imaging geometry of PCS vision sensors. New algebraic results on the multiple-view geometry for the case of static and moving cameras (multi-view PCS) are introduced. On the basis of these new results, we addressed both the image-based camera localization and mirror calibration problems. Extensive simulation and experimental results have been presented in order to illustrate the theory and show the applicability of the proposed algorithms in real scenarios.

The extension of our results to multiple camera networks is a subject of future research. Future investigations will also focus on the 3D scene reconstruction problem and will show the connection between PCS sensors and the trifocal geometry [9].

Acknowledgements. The authors are grateful to Francesco Chinello for his help in the preparation of the experimental setup.

References

[1] Armangué, X., Salvi, J.: Overall View Regarding Fundamental Matrix Estimation. Image Vision Comput. 21(2), 205–220 (2003)

[2] Bouguet, J.: Camera Calibration Toolbox for Matlab (2007), `http://www.vision.caltech.edu/bouguetj`

[3] Boult, T., Nayar, S.: A Handy Stereo Camera. University of Columbia and Colorado (2000), `http://vast.uccs.edu/~tboult/VSAM.html/#stereo`

[4] Geyer, C., Daniilidis, K.: Catadioptric Camera Calibration. In: Proc. 7th IEEE Int. Conf. Comp. Vis., vol. 1, pp. 398–404 (1999)

[5] Gluckman, J., Nayar, S.: Planar Catadioptric Stereo: Geometry and Calibration. In: Proc. IEEE Conf. Comp. Vis. and Pattern Rec., vol. 1, pp. 22–28 (1999)

[6] Gluckman, J., Nayar, S.: Catadioptric Stereo using Planar Mirrors. Int. J. Comput. Vision 44(1), 65–79 (2001)
[7] Gluckman, J., Nayar, S.: Rectified Catadioptric Stereo Sensors. IEEE Trans. Pattern Anal. Mach. Intell. 24(2), 224–236 (2002)
[8] Goshtasby, A., Gruver, W.: Design of a Single-Lens Stereo Camera System. Pattern Recogn. 26(6), 923–937 (1993)
[9] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2004)
[10] Hong, W., Yang, A.Y., Huang, K., Ma, Y.: On Symmetry and Multiple-View Geometry: Structure, Pose, and Calibration from a Single Image. Int. J. Comput. Vision 60(3), 241–265 (2004)
[11] Konrad, J.: View Reconstruction for 3-D Video Entertainment: Issues, Algorithms and Applications. In: Proc. IEEE Int. Conf. Image Proc. and Appl., vol. 1, pp. 8–12 (1999)
[12] Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 20(2), 91–110 (2004)
[13] Ma, Y., Soatto, S., Kosecká, J., Sastry, S.: An Invitation to 3-D Vision: From Images to Geometric Models. Interdisciplinary Applied Mathematics. Springer, Heidelberg (2003)
[14] Mariottini, G., Prattichizzo, D.: EGT for Multiple View Geometry and Visual Servoing: Robotics Vision with Pinhole and Panoramic Cameras. IEEE Robot. Autom. Mag. 12(4), 26–39 (2005)
[15] Mariottini, G., Prattichizzo, D.: Image-based Visual Servoing with Central Catadioptric Camera. Int. J. Robot. Res. 27(1), 41–57 (2008)
[16] Mariottini, G., Scheggi, S., Morbidi, F., Prattichizzo, D.: Planar Catadioptric Stereo: Single and Multi-View Geometry for Calibration and Localization. In: Proc. IEEE Int. Conf. Robot. Automat, pp. 1510–1515 (2009)
[17] Mitsumoto, H., Tamura, S., Okazaki, K., Kajimi, N., Fukui, Y.: 3-D Reconstruction using Mirror Images based on a Plane Symmetry Recovering Method. IEEE Trans. Pattern Anal. Mach. Intell. 14(9), 941–946 (1992)
[18] Nayar, S.: Catadioptric Omnidirectional Camera. In: Proc. IEEE Conf. Comp. Vis. and Pattern Rec., pp. 482–488 (1997)
[19] Nene, S., Nayar, S.: Stereo with Mirrors. In: Proc. 6th IEEE Int. Conf. Comp. Vis., pp. 1087–1094 (1998)
[20] Pachidis, T., Lygouras, J.: A Pseudo Stereo Vision System as a Sensor for Real-Time Path Control of a Robot. In: Proc. 19th IEEE Conf. Instr. and Measurem. Tech., vol. 2, pp. 1589–1594 (2002)
[21] Svoboda, T., Pajdla, T.: Epipolar Geometry for Central Catadioptric Cameras. Int. J. Comput. Vision 49(1), 23–37 (2002)
[22] Wu, H.H.: Rectification of Stereoscopic Video for Planar Catadioptric Stereo Systems. IEEE Trans. Circ. Syst. Video Tech. 17(6), 686–698 (2007)

Chapter 2
Empirical Characterization of Convergence Properties for Kernel-based Visual Servoing

John P. Swensen, Vinutha Kallem, and Noah J. Cowan

Abstract. Visual servoing typically involves separate feature tracking and control processes. Feature tracking remains an art, and is generally treated as independent of the underlying controller. Kernel-based visual servoing (KBVS) is a categorically different approach that eliminates explicit feature tracking. This chapter presents an experimental assessment of the convergence properties (domain of attraction and steady-state error) of the proposed approach. Using smooth weighting functions (the kernels) and Lyapunov theory, we analyze the controllers as they act on images acquired in controlled environments. We ascertain the domain of attraction by finding the largest positive invariant set of the Lyapunov function, inside which its time derivative is negative definite. Our experiments show that KBVS attains a maximum pixel error of one pixel and is commonly on the order of one tenth of a pixel.

2.1 Featureless Visual Servoing

Typically, visual servoing involves tracking image features and controlling a robot based on the motions of these image features. Usually this involves tuning the feature tracking algorithm and controller independently, with no clear notion of how to co-optimize tracking and control. KBVS is distinguished from traditional visual servoing techniques in two respects, namely the lack of explicit feature tracking at each image frame and the inherent combination of tracking and control. These

John P. Swensen and Noah J. Cowan
Department of Mechanical Engineering, Johns Hopkins University,
3400 North Charles Street, Baltimore, MD 21218, USA
e-mail: {jpswensen,ncowan}@jhu.edu

Vinutha Kallem
GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street,
Philadelphia, PA 19104, USA
e-mail: vkallem@seas.upenn.edu

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 23–38.
springerlink.com

characteristics may provide a benefit over traditional position-based and image-based visual servoing [10], 2 1/2 D visual servoing [2], and other advanced visual servo algorithms [1].

"Featureless" visual servoing methods including KBVS take advantage of the rich set of visual data present in images without reducing the image to feature points. This reduces computation because extracting image features usually requires more computation than the image measurements used in featureless methods. Also, this confers robustness to region or feature occlusions in the images.

Nayar *et al.* [15] present one of the earliest works in featureless visual servoing. The authors select patches of the goal image that form a high dimensional measurement. They then generate a training data set of images acquired at known robot poses. The leading principal components of the training images form a low dimensional approximation of the so-called appearance manifold. Images during control are projected onto this manifold and control is performed on this low dimensional space. The sampling of the camera workspace becomes prohibitively time consuming as the number of degrees of freedom increases and has to be evaluated for each scene. Deguchi [6] extends the method to 6 degrees of freedom and automates patch selection to improve the Jacobian. These papers do not address formal guarantees of convergence.

More recently, Tahri and Chaumette [16] use moments for determining camera motions with respect to a goal. They compute the Jacobian relating changes in low-order image moments to the velocity of the moving camera. The image moments provide a similar measurement as the KBVS sampling kernel in that they provide a scalar measurement of the entire image for each type of moment calculation. Formally, with an appropriately designed family of kernels, it may be possible to conceive of moment-based visual servoing as a special case of KBVS. The advantages of KBVS are that selection of kernels with compact (finite) support will minimize edge effects, and moreover the KBVS approach embraces gray scale images. However, the image moment solution provides for 6 degree of freedom visual servoing, a problem not yet solved for KBVS.

A related result by Collewet *et al.* [3] notes that for a static scene, the time-varying image is simply a transformed version of the original image, assuming no occlusions. They develop the Jacobian relating the motion of every pixel in the image to the motion of the camera, thus allowing derivatives of the image signal to be calculated directly. They then use the time derivative of the image to minimize a cost function, which in similar to the Lyapunov function as discussed later in this chapter, if the image is a continuous signal. This approach is analogous to placing a kernel at every pixel.

KBVS presents a new kind of featureless visual servoing and this chapter presents the first in-depth empirical evaluation of KBVS. Section 2.2 presents the KBVS algorithm and its conceptual and theoretical underpinnings. Section 2.3 describes the experiments conducted to evaluate the method and characterize the convergence properties for several degrees of freedom. We conclude by discussing the outstanding issues that need to be addressed to make KBVS robust and to rationalize kernel selection.

2.2 Kernel-based Visual Servoing

To aid the reader in understanding the basics of KBVS, we describe the details of the algorithm in a simple two dimensional case [12] and refer the reader to Kallem *et al.* [13] for the details of the SO(3) case, roll about the optical axis, translation along the optical axis, and combined motion derivations.

The kernel-projected value in KBVS, derived from the kernel-based tracking literature [4, 9, 8, 7, 17], is a weighted measurement of an image based on the a sampling kernel. A Lyapunov function is formed from the vector of kernel-projected values and the KBVS control input ensures Lyapunov stability, as shown below. A key concept in the demonstration of stability is the equivalence of the kernel-projected value under a change of coordinates: ideally there is no difference between the kernel-projected value under a transformation of the image or the inverse transformation of the kernel. In the two dimensional case discussed below, this transformation is simply translation parallel to the image plane of the camera.

Throughout the analytical derivation of KBVS, several assumptions are made about the scene and the signals that appear in the computation. First, we assume that the image plane is continuous (rather than discrete) and infinite, namely the image plane is a copy of $\mathbb{R}^2$. Second, we assume the scene to be planar. Finally, we require that pixels are constantly illuminated across all image frames.

We assume the robot, as seen in Figure 2.1, is the kinematic plant

$$\dot{\mathbf{x}} = \mathbf{u} \tag{2.1}$$

where the configuration of the robot is described as $\mathbf{x} = [x, y]^T$ and $\mathbf{u} \in \mathbb{R}^2$ drives the robot parallel to the image plane of the camera. Let the image be represented as a

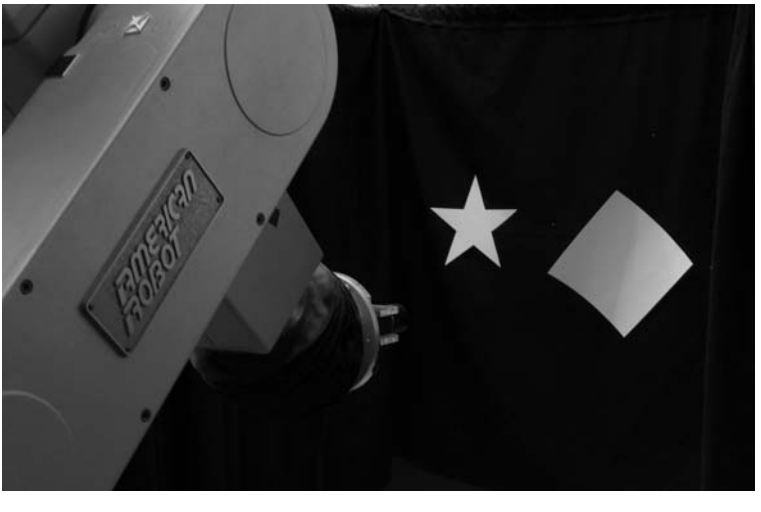


Fig. 2.1 The robot used for the experiments done in this chapter

signal $I(\mathbf{w},\mathbf{x}(t)) \in \mathbb{R}$. The spatial indexing parameter of the 2D image is $\mathbf{w} \in \mathscr{I} = \mathbb{R}^2$. The signal is represented as a function of the spatial indexing and the position which is a function of time. Although one might expect to see the signal written only as a function of time, $I(\mathbf{w},t)$, we will be more explicit and write it as a function of the robot configuration which is in turn a function of time, $I(\mathbf{w},\mathbf{x}(t))$. For the sake of notational simplicity and without loss of generality we assume that the image plane is a unit distance away from the scene.

Using the modified notation, the signal at any translated position is related to the signal at the goal

$$I(\mathbf{w},\mathbf{x}(t)) = I(\mathbf{w}-\mathbf{x}(t),\mathbf{0}) = I_0(\mathbf{w}-\mathbf{x}(t)). \tag{2.2}$$

In the two dimensional case, the kernel-projected value is a function $\xi : \mathscr{I} \to \mathbb{R}^n$ where

$$\begin{aligned} \xi &= \int_{\mathscr{I}} K(\mathbf{w}) I(\mathbf{w},t) d\mathbf{w} \\ &= \int_{\mathscr{I}} K(\mathbf{w}) I_0(\mathbf{w}-\mathbf{x}(t)) d\mathbf{w}, \end{aligned} \tag{2.3}$$

and the kernel-projected value at the goal is $\xi_0 = \xi$ at x_0. The dimension n refers to the number of kernels being used such that the kernels represent a function $K : \mathbb{R}^2 \to \mathbb{R}^n$. Note that using the change of coordinates $\bar{\mathbf{w}} = \mathbf{w} - \mathbf{x}(t)$, (2.3) can be written as

$$\xi = \int_{\mathscr{I}} K(\bar{\mathbf{w}} + \mathbf{x}(t)) I_0(\bar{\mathbf{w}}) d\bar{\mathbf{w}}. \tag{2.4}$$

(2.3) and (2.4) show that the the kernel-projected value is the same for the fixed kernel with an image taken after the camera has been moved and a kernel shifted in the opposite direction with the goal image. This can be seen pictorially in Figure 2.2 for a one-dimensional signal and kernel. The change of coordinates in (2.4) will be used later to allow us to take time derivatives of the kernel-projected value: we can easily design kernels with known and analytically determined derivatives, whereas the time derivatives of the images are unknown.

The goal of the KBVS method is to find a control input $\mathbf{u}$, a function of ξ, to move the robot such that

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}_0.$$

To find such a control input, we consider

$$V = \frac{1}{2}(\xi - \xi_0)^T P (\xi - \xi_0) \tag{2.5}$$

as a Lyapunov function candidate, where P is any positive $n \times n$ matrix. To choose the control input, $\mathbf{u}$, we analyze the time derivative of the Lyapunov function using the shifted kernel representation of the kernel-projected value from (2.4):

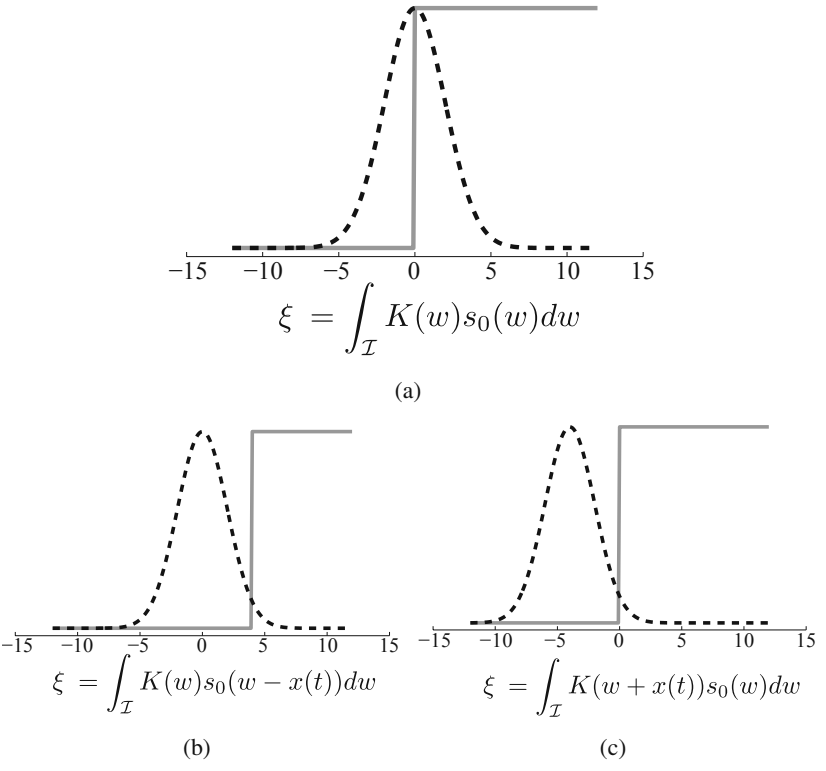


Fig. 2.2 Demonstration of the equivalence of the kernel-projected value whether moving the camera or moving the kernel: (a) a static kernel and the signal at the goal; (b) a static kernel and a shifted signal; and (c) a shifted kernel and the signal at the goal

$$\begin{aligned}\dot{V} &= (\xi - \xi_0)^T P \dot{\xi} \\ &= (\xi - \xi_0)^T P \frac{\partial \xi}{\partial \mathbf{x}} \dot{\mathbf{x}} \\ &= (\xi - \xi_0)^T P \left[\underbrace{\int_{\mathscr{I}} \frac{\partial K(\bar{\mathbf{w}} + \mathbf{x}(t))}{\partial \mathbf{x}} I_0(\bar{\mathbf{w}}, \mathbf{0}) d\mathbf{w}}_{\text{Jacobian } J} \right] \mathbf{u}.\end{aligned} \tag{2.6}$$

After reversing the change of coordinates, (2.6) can be written as

$$\begin{aligned}\dot{V} &= (\xi - \xi_0)^T P \left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I_0(\mathbf{w} - \mathbf{x}(t), \mathbf{0}) d\mathbf{w} \right] \mathbf{u} \\ &= (\xi - \xi_0)^T P \left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I(\mathbf{w}, \mathbf{x}(t)) d\mathbf{w} \right] \mathbf{u}.\end{aligned} \tag{2.7}$$

If we choose the control input

$$\mathbf{u} = -\left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I(\mathbf{w}, \mathbf{x}(t)) d\mathbf{w}\right]^T P^T (\xi - \xi_0) \tag{2.8}$$

the time derivative of V is given by

$$\dot{V} = -\left\| (\xi - \xi_0)^T P \left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I(\mathbf{w}, \mathbf{x}(t)) d\mathbf{w}\right] \right\|^2 \leq 0. \tag{2.9}$$

We now have a Lyapunov function that is positive definite in ξ by construction and have shown a control input $\mathbf{u}$ that ensures its time derivative is negative semidefinite in ξ. Asymptotic stability to the goal configuration, however, must be shown in $\mathbf{x}(t)$ (assuming without loss of generality that $\mathbf{x}_0 = 0$). That is, we must show that both V and $\dot{V}$ are positive definite and negative semidefinite in $\mathbf{x}(t)$, respectively. To see this locally, we look at the first order Taylor series expansion of ξ about the point ξ_0, as done for classical visual servoing in [5]:

$$\begin{aligned} \xi &= \xi_0 + \frac{\partial \xi}{\partial \mathbf{x}} \mathbf{x}(t) + O(\mathbf{x}^2) \\ \xi - \xi_0 &= \frac{\partial \xi}{\partial \mathbf{x}} \mathbf{x}(t) + O(\mathbf{x}^2) \\ \xi - \xi_0 &= J\mathbf{x}(t) + O(\mathbf{x}^2). \end{aligned} \tag{2.10}$$

Then, inserting (2.10) into (2.5) and (2.6) we achieve the desired definiteness in $\mathbf{x}(t)$ under certain conditions on the Jacobian matrix J:

$$V = \frac{1}{2} \mathbf{x}(t)^T J^T P J \mathbf{x}(t) + O(\mathbf{x}^3), \quad Q = J^T P J, \tag{2.11}$$

and

$$\dot{V} = -\mathbf{x}(t)^T Q Q^T \mathbf{x}(t) + O(\mathbf{x}^3). \tag{2.12}$$

If the Jacobian matrix $J \in \mathbb{R}^{n \times p}$ is full column rank, where n is the number of kernels and p is the dimension of $\mathbf{x}(t)$, then Q will be a full rank $p \times p$ matrix. From (2.11), (2.12), and a full rank assumption for Q, we can conclude that V is positive definite and $\dot{V}$ is negative definite in some neighborhood of the goal with $V = 0$ and $\dot{V} = 0$ at the goal. The rank condition on Q also clearly sets forth a necessary condition on the number of kernels required; namely that there must be at least as many kernel measurements as there are degrees of freedom.

Although we have only shown the method for the simple 2D case, the reader can infer how it extends to other motions given an appropriate change of coordinates based on the relationship between camera motion and image transformation. Several motions are worked out in great detail in our previous work [12, 11, 13]. These include 2D translation, depth, roll, rigid body rotations, and some combinations of these.

Considering the computation and assumption from above, to practically implement a KBVS controller we must relax several of the assumptions made about the scene and signals. First, the image plane is not continuous. Our kernel-projected value then becomes a discretized summation, rather than a continuous integral:

$$\xi = \int_{\mathscr{I}} K(\mathbf{w}) I(\mathbf{w}(t)) d\mathbf{w} \approx \sum_{w} \bar{K}[w] \bar{I}[w(t)]. \tag{2.13}$$

Additionally, the image plane in not infinite. In (2.13), the domain of the integration would be infinite whereas the domain of the summation extends to the boundaries of the image. We work around this problem by choosing kernels with finite support. That is, the kernels are selected such that the weighting is zero at the boundaries of the image. This allows us to truncate the integration, as the integral over the domain outside the image boundaries will essentially be zero.

Another limitation is the assumption of a planar scene. This is necessary to avoid issues of parallax between near and far objects in the scene. Scene parallax could significantly affect the size of the domain of attraction. In our experimental configurations we have constructed a planar scene. In real world implementations, one can ensure that the kernels and their finite support are wholly contained within an approximately planar object.

Because we are making a weighted measurement on the intensities of each pixel individually, we need to assume that each point in space is providing the same measurement as it moves across the pixels of the camera. This is also known as the brightness constancy constraint. To avoid problems of varying lighting, we normalize the image at each time step to the maximum pixel value.

The first aim of the following sections is to verify the KBVS method empirically, taking into consideration the fact that we are violating the assumptions of the analytical solution. During experimentation, we are using the workarounds for each of the nonideal issues as described above. Additionally, the analysis above only determined that the KBVS control input produces an asymptotically stable system in a neighborhood of the goal configuration, but gave no insight into the size of the neighborhood. The second aim of the empirical validation is to characterize the domain of attraction for each of the degrees of freedom discussed above.

2.3 Empirical Validation

Empirical validation of KBVS is a two step process. The first is to characterize the domain of attraction for a kernel and scene combination. As discussed during the analytical derivation, there are scenes and/or kernel combinations that could result in either a poor domain of attraction or instability. After finding a good set of kernels for the scene, per the Jacobian rank condition, and having ascertained the expected domain of attraction, the second step is to use the expected domain of attraction for initial conditions to a set of experiments.

Subsection 2.3.1 discusses the domain of attraction for several of the combinations of degrees of freedom for which we have an analytical solution as indicated in

Section 2.1 and calculated in [12, 13]. In order to visualize the domain of attraction, we show the level sets of V in most cases and a level surface plus three orthogonal slices of the level surface that pass through the origin for the $\mathrm{SO}(3)$ case. In each figure we also plot the sign of $\dot{V}$ at each point.

Subsection 2.3.2 then uses the results of Section 2.3.1 to show experimental results for a variety of the combinations of the degrees of freedom in the controlled environments. We conduct 50 trials for each type of robot motion by going to the edge of the domain of attraction as determined in Section 2.3.1.

2.3.1 Analysis of the Domain of Attraction

To find the domain of attraction experimentally, we perform the following sequence for each type of robot motion:

1. gather images on an evenly space grid of the camera workspace;
2. compute the value of the Lyapunov function, V, at each position;
3. compute the control input, $\mathbf{u}$, at each position as described in Section 2.2;
4. compute the gradient of the Lyapunov function, ∇V, using a finite central difference approximation at each position;
5. compute the time derivative of the Lyapunov function using the chain rule

$$\dot{V} = \nabla V \dot{\mathbf{x}} = \nabla V \mathbf{u}; \quad \text{and} \tag{2.14}$$

6. search for the level set of V which is homeomorphic to S^n with the largest interior and for which $\dot{V} < 0$ for every point on the interior except the goal position.

By the theory of Lyapunov stability[14], we can conclude that the volume described by the interior of the level set of the Lyapunov function as described above is the domain of attraction for the asymptotically stable system.

2.3.1.1 2D

Figure 3(b) shows the analysis of the domain of attraction for the goal scene and kernels shown in Figure 3(a). The chosen kernels are placed around the star object and provide for a domain of attraction that approaches the joint limits of the robot for planar motions. Although not seen in this figure, kernel selection often results in a Lyapunov bowl that is not symmetrically shaped. The positive definite matrix P from (2.5) can be engineered to change the shape of the Lyapunov function and provide faster convergence in all directions.

In contrast, Figures 3(c) and 3(d) show how a poor selection of kernels unintentionally resulted in a second subset of the domain of the Lyapunov function to which the system would converge. Similar to the successful set of kernels, these kernels were placed at different locations around the star object near the center of the goal image. The combination of scene and kernels resulted in an invariant region for which the robot motion was stable to some point, but not asymptotically stable

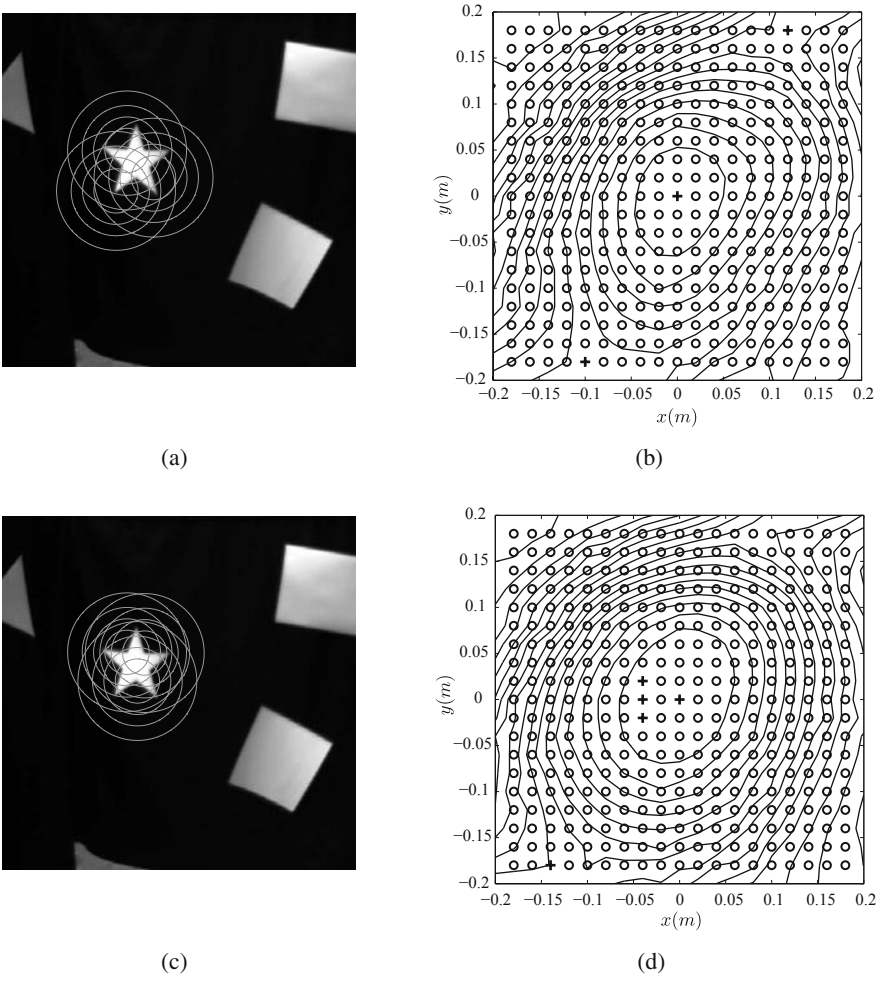
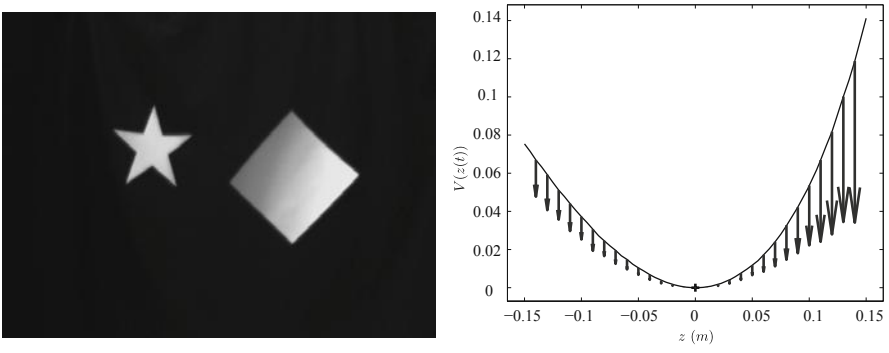


Fig. 2.3 An analysis of the domain of attraction for good and bad kernel selection: (a) the goal image and the contours of the good kernels; (b) the level curves of V for the good kernels. The $+$ symbols indicate locations at which $\dot{V} \geq 0$ and the $\circ$ symbols indicate locations where $\dot{V} < 0$; (c) the goal image and the contours of the bad kernels; and (d) the level curves of V for the bad kernel with the associated marker for $\dot{V}$

to the goal for all the initial condition within the invariant region. This illustrates the need for a principled approach to kernel selection.

2.3.1.2 Depth

The domain of attraction for the depth motion was as large as the joint limits of the robot allowed. That is, the Lyapunov function consisted of a convex quadratic



(a) (b)

Fig. 2.4 An analysis of the domain of attraction for depth motion: (a) the goal image and the contours of the kernels used; and (b) the plot of V. The arrow indicate the value of $\dot{V}$ at each depth

function of the camera depth along the optical axis for which $\dot{V}$ was negative everywhere except the goal location and zero at the goal, as seen in Figure 2.4.

2.3.1.3 SO(3)

Figure 2.5 shows the analysis of the domain of attraction for the set of kernels shown in Figure 2.6. To determine the domain of attraction, we determined all the orientations of the robot for which our empirical computation of the time derivative of the Lyapunov function was positive. We then start at the level surface $V = c$ for c very large and slowly decrease c until arriving at a level surface where $\dot{V} < 0$ for every point in its interior except at the goal. The volume defined by the interior of the level surface defines the domain of attraction for this 3 degrees of freedom system. Figure 5(a) shows this level surface and nearby points where $\dot{V} > 0$. Figures 5(b), 5(c) and 5(d) show orthogonal slices of the Lyapunov function through the origin. It becomes much easier to visualize the magnitude and shape of the domain of attraction using these slices, taking into account that these may give a inaccurate representation for strangely shaped level surfaces of V.

Figure 2.6 shows the kernels used for determining the domain of attraction and for the experiments shown later. It is often useful to show the kernel overlayed on top of the image at the goal position to give an idea of how the kernel-projected values are generated from the image at the goal. It is important to note that a change in the kernels, whether location or covariance for our Gaussian kernels, carries with it an implicit change in the domain of attraction.

2.3.2 Experimental Results

To validate the domains of attraction determined above, we conducted 50 trials for each of the degrees of freedom analyzed in Section 2.3.1. We randomly set the

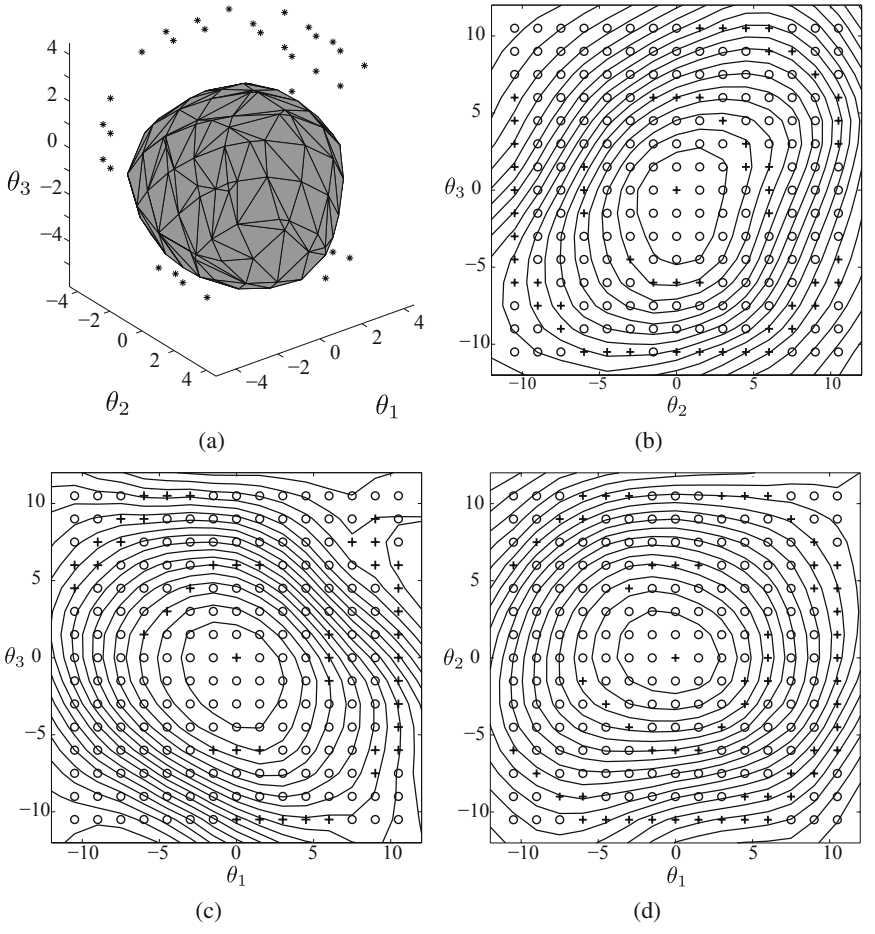
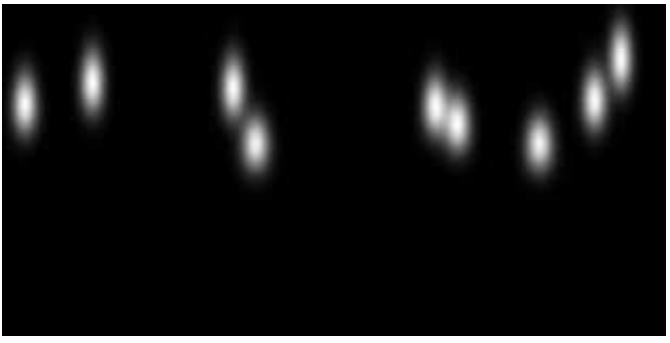


Fig. 2.5 The domain of attraction for rigid body rotations: (a) the volume representing the domain of attraction with nearby points where $\dot{V} > 0$; (b) the slice through the level sets of V through the origin and aligned with the θ_1 axis; (c) the slice aligned with the θ_2 axis; and (d) the slice aligned with the θ_3 axis. The $+$ symbols indicate locations at which $\dot{V} \geq 0$ and the $\circ$ symbols indicate locations where $\dot{V} < 0$

initial displacement of the robot to be near the empirically determined limits of the domain of attraction, except for the case of the rigid body rotations as discussed later. Error measurements were computed using the forward kinematics of the robot to determine the goal pose of the robot, the initial displacement, and the final pose of the robot.

Figure 2.7 shows the camera trajectories of the robot in the (x,y) plane for the 50 experimental trials of 2D motion parallel to the image plane. The statistics concerning average error, standard deviation of error, and maximum and minimum errors is given in Table 2.1. A point of interest is the minimum final error of 90 microns.



(a)

(b)

Fig. 2.6 Useful visualizations for placing kernels. These provide insight into how the kernels interact with an image: (a) the sum of the kernels; and (b) the kernels overlayed on the image at the goal location

This represents an error of 2-5 encoder counts for each of the joint variables of the robot. While the maximum error was 1.5 orders of magnitude larger, there were many trials that were near this minimum error.

Figure 2.8 shows the camera trajectories of the robot along the optical axis for the 50 experimental trials of depth motion. The corresponding error statistics are shown in Table 2.2.

Figure 2.9 shows the rotations about the focus of the parabolic mirror for the 50 experimental trials of rigid body rotations. The error statistics for the angles are found in Table 2.3. The determination of the domain of attraction for the rigid body rotations indicated that the maximum allowable move would be approximately $\pm 5^\circ$ in each of the angles. However, experimentation showed that the system rarely failed for initial displacements of approximately $\pm 9^\circ$. Upon closer inspection of Figure 2.5, it is evident that there is a region of the volume near the 5° radius and the 9° radius where $\dot{V}$ is positive. Figure 2.9 shows one failed trial whose initial

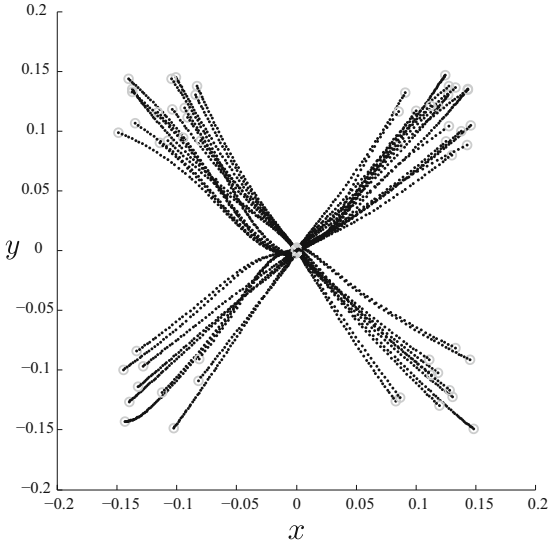


Fig. 2.7 Convergence results for the 50 trials of 2D translations orthogonal to the optical axis

Table 2.1 Error measurements for the fifty 2D experiments

	Average Error	σ Error	Max Error	Min Error
x	0.3347 mm	0.2472 mm	1.0527 mm	4.989×10^{-3} mm
y	0.6857 mm	0.5792 mm	2.4527 mm	9.6609×10^{-3} mm
$\sqrt{x^2+y^2}$	0.9439 mm	0.5267 mm	2.4598 mm	89.5382×10^{-3} mm

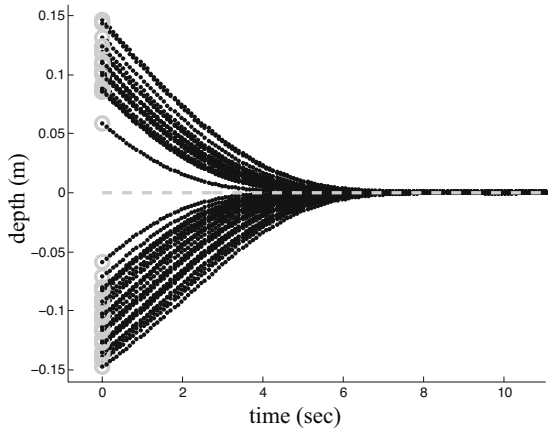


Fig. 2.8 Convergence results for the 50 trials of translation parallel to the optical axis

displacement was at the 9° radius, which was not included in the calculation of error statistics.

2.3.3 Final Error in Pixels

Although the final errors in configuration space given in Section 2.3.2 seem more that adequate, a more insightful measure of efficacy is the final error in pixels. Because perspective cameras were used, the distance of the objects from the camera play a significant role in how the objects move, and consequently, the magnitude of change in the kernel projected value under camera motions. The number of pixels per unit of motion was determined using the set of collected images from Section 2.3.1 by following a point during motion in each of the degrees of freedom.

For the 2D experiments, the plane containing the object was at a distance such that the relationship between robot motion and image motion was approximately 2 mm/pixel. From Table 2.1 we see that the maximum error was on the order of a pixel and the average error was approximately a half of a pixel.

For the depth experiments, the plane containing the object was at a distance such that the relationship between the robot depth motion and the image motions was approximately 4 mm/pixel. From Table 2.2 we see that the maximum error was about one-third of a pixel and the average error was approximately one-tenth of a pixel.

For the rigid body rotation experiments, the results were very promising in terms of pixel error. Because the image is the result of a spherical projections, the pixel error changes depending on whether it is measured near the center of the image or

Table 2.2 Error measurements for the fifty depth experiments

	Average Error	σ Error	Max Error	Min Error
z	0.3708 mm	0.2864 mm	1.2163 mm	0.0203 mm

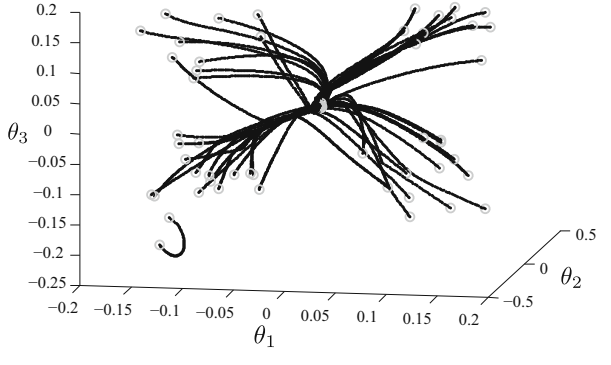


Fig. 2.9 Convergence results for the 50 trials of rigid body rotations

Table 2.3 Error measurements for the fifty rigid body rotation experiments

	Average Error	σ Error	Max Error	Min Error
θ_1	0.0391°	0.0257°	0.1277°	2.2192×10^{-6} °
θ_2	0.0417°	0.0314°	0.1175°	6.18×10^{-4} °
θ_3	0.0474°	0.0463°	0.2649°	0.0017°
$\sqrt{\theta_1^2 + \theta_2^2 + \theta_3^2}$	0.0853°	0.0446°	0.2781°	0.0151°

near the periphery. Since most of our objects and kernels were between these two extremes, the pixel error was calculated based on pixel motions in this intermediate region. The structure containing the objects was at a distance such that the relationship between the rotations about the focus of the parabolic mirror and the image motions was approximately 0.6°/pixel. Referring to Table 2.3 we see that the maximum error was about one-third of a pixel and the average error was about one-fifth of a pixel.

2.4 Conclusions

The results of the analysis in Section 2.3.1 and experimentation in Section 2.3.2 indicate that relaxing the assumptions required for the analytical treatment of KBVS does not hamper experimental performance. We achieved adequate convergence rate and domain of attraction. Moreover, the KBVS results in very low steady-state error, despite our current *ad hoc* approach to kernel selection.

Robust kernel selection for unstructured environments is an area for future research. Figure 2.3 demonstrated how slight variations in kernel selection could cause the robot to fail to converge to the goal image. Even though both sets of kernel satisfied the rank requirement for the Jacobian between $\dot{\xi}$ and $\dot{\mathbf{x}}$, one choice is clearly superior with respect to the size of the domain of attraction. The ability to optimize kernel selection to maximize the domain of attraction using only the image at the goal is the natural next step.

References

[1] Chaumette, F., Hutchinson, S.: Visual servo control, part ii: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)

[2] Chaumette, F., Malis, E.: 2 1/2 D visual servoing: a possible solution to improving image-based and position based visual servoings. In: IEEE International Conference on Robotics and Automation (2000)

[3] Collewet, C., Marchand, E., Chaumette, F.: Visual servoing set free from image processing. In: IEEE International Conference on Robotics and Automation, pp. 81–86 (2008)

[4] Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5), 564–575 (2003)

[5] Cowan, N.J.: Vision-based control via navigation functions. Ph.D. thesis, University of Michigan (2001)
[6] Deguchi, K.: A direct interpretation of dynamic images with camera and object motions for vision guided robot control. International Journal of Computer Vision 37(1), 7–20 (2000)
[7] Dewan, M., Hager, G.: Towards optimal kernel-based tracking. In: Computer Vision and Pattern Recognition, vol. 1, pp. 618–625 (2006)
[8] Fan, Z., Wu, Y., Yang, M.: Multiple collaborative kernel tracking. In: Computer Vision and Pattern Recognition (2005)
[9] Hager, G.D., Dewan, M., Stewart, C.V.: Multiple kernel tracking with SSD. In: Computer Vision and Pattern Recognition, vol. 1, pp. 790–797 (2004)
[10] Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12(5), 651–670 (1996)
[11] Kallem, V.: Vision-based control on lie groups with application to needle steering. Ph.D. thesis. Johns Hopkins University (2008)
[12] Kallem, V., Dewan, M., Swensen, J.P., Hager, G.D., Cowan, N.J.: Kernel-based visual servoing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, pp. 1975–1980 (2007)
[13] Kallem, V., Dewan, M., Swensen, J.P., Hager, G.D., Cowan, N.J.: Kernel-based visual servoing: featureless control using spatial sampling functions (in preparation)
[14] Khalil, H.K.: Nonlinear Systems, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
[15] Nayar, S.K., Nene, S.A., Murase, H.: Subspace methods for robot vision. IEEE Transactions on Robotics and Automation 12(5), 750–758 (1996)
[16] Tahri, O., Chaumette, F.: Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics 21(6), 1116–1127 (2005)
[17] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4) (2006)

Chapter 3
High-speed Visual Feedback Control for Grasping and Manipulation

Akio Namiki, Taku Senoo, Satoru Mizusawa, and Masatoshi Ishikawa

Abstract. Recently, several types of high-speed vision in which the sampling rate is more than 1 kHz have been developed. In the high-speed vision system, not only sensing but also processing is achieved at high-speed, and it improves the performance of visual servoing. In this chapter, we described some examples of visual feedback control methods in our high-speed manipulation system. First, a hybrid trajectory generator using visual information is proposed for a batting task. In the method, the manipulator trajectory defined as a polynomial function of time is modified by visual information directly. Next, a visual servoing control with a passive joint problem is proposed for tool manipulation. Experimental results are shown.

3.1 Introduction

In general-purpose dexterous manipulation systems, it is necessary to manipulate many types of objects. In order to recognize any type of object, vision is one of the most important sensations for manipulation. In usual vision systems such as charge-coupled device (CCD) camera, the sampling rate is 30 Hz which is the same rate of human visual recognition. The rate is not enough to recognize an object moving at high-speed. A human solves this problem by his excellent ability to predict the future, however, its result is not always reliable.

Akio Namiki
Department of Mechanical Engineering, Chiba University,
1-33 Yayoi-cho, Inage, Chiba, Japan
e-mail: namiki@faculty.chiba-u.jp

Taku Senoo, Satoru Mizusawa, and Masatoshi Ishikawa
Department of Information Physics and Computing, University of Tokyo,
7-3-1 Hongo, Bunkyo, Tokyo, Japan
e-mail: {taku_senoo,satoru_mizusawa}@ipc.i.u-tokyo.ac.jp,
masatoshi_ishikawa@ipc.i.u-tokyo.ac.jp

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 39–53.
springerlink.com

Recently, several types of high-speed vision in which the sampling rate is more than 1 kHz have been developed [4, 6]. In the high-speed vision system, not only sensing but also processing is achieved at high-speed. It is the main difference of the high-speed vision from a high-speed camera which is used for recording the video in slow motion. In order to achieve rapid processing, mostly a type of parallel processing architecture are used.

There are several advantages of high-speed vision in manipulation tasks. First, the high-speed vision is useful to recognize an object moving at high-speed. For a high-speed vision system, the object moving at high-speed seems to be stopping. For this reason, a complex prediction process is not needed. Because the transition during one frame is small, the scanning and exploration process in the image plane become also useless. The high-speed vision is also useful to recognize a flexible transformable object such as rope or cloths. Secondly, a high-speed vision at a rate of 1 kHz can be included to a feedback control loop directly. In robot manipulation systems, the sampling rates of other sensors such as an optical encoder and a strain gage are about 1 kHz. A high-speed vision system can be used with the other sensors simultaneously. Also, because the delay is small, a high-gain visual feedback can be achieved. These advantages show that a high-speed vision improves the performance of visual servoing. As a result, a visual control only based on the appearance without the precise calibration can be achieved.

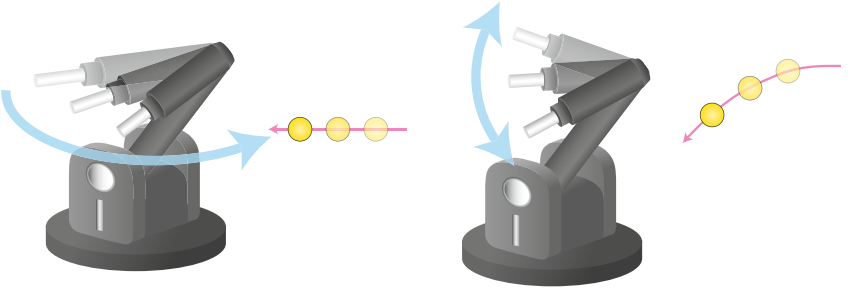
Our research group have developed a manipulation system with high-speed vision, and by using the system several types of high-speed manipulation have been achieved: dynamic regrasping [1], stick spinning [3], and rope knotting [11]. In this chapter, we propose a high-speed batting task [9, 10] and a tool manipulation task [5] as examples of the visual feedback control using high-speed vision.

3.2 High-speed Batting

A human batting task in the baseball game is a task which needs high-level visual control ability. In general, it takes a human more than 0.2 s to act based on visual information, and it takes a ball only 0.1 s that to reach a catcher's mitt. For this reason, a human must predict the ball trajectory using some other information such as the pitcher's motion. However, it is difficult to predict it perfectly, and he often misses. If we use a high-speed vision, we can realize a more reliable batting system. We proposed a hybrid trajectory generator as a motion strategy for a high-speed robot system and achieved the batting task [9, 10]. This trajectory consists of both a swing motion and a tracking motion.

3.2.1 Hybrid Trajectory Generator

The batting motion consists of two motions as shown in Figure 3.1: one is the swing motion, and the other is the hitting motion. The swing motion is necessary to hit a ball hard, and it should be controlled so that its speed approaches the maximum. In order to achieve a smooth high-speed motion, this motion should be represented as



(a) SW mode (b) HT mode

Fig. 3.1 Hybrid trajectory [10]

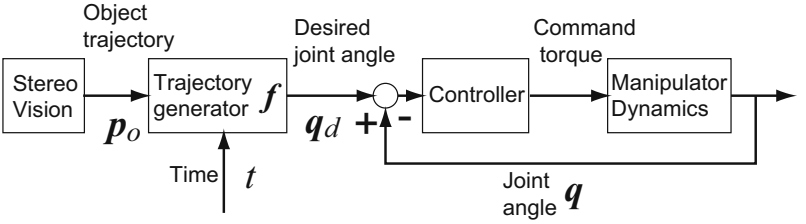


Fig. 3.2 Block diagram

the function of the time variable t, and it makes easy to design a controller. We call this motion the "SW" mode. Then, the hitting motion is the tracking of the change of a ball trajectory, and this motion is necessary to hit a ball precisely. This motion can be achieved by mapping the trajectory of the manipulator to the ball position directly. This motion is represented as the function of the object position $\mathbf{p}_0$. We call this motion the "HT" mode.

By integrating the two modes, the desired trajectory of the joint angle vector of the manipulator $\mathbf{q}_d \in \mathbb{R}^n$ is defined as

$$\mathbf{q}_d = \mathbf{f}(\mathbf{p}_0, t) \in \mathbb{R}^n \,, \tag{3.1}$$

where $\mathbf{f}$ is an appropriate nonlinear function, and n is the number of joints. The joint angle vector $\mathbf{q}$ is controlled so as to track $\mathbf{q}_d$ using an appropriate controller. Figure 3.2 shows the block diagram.

3.2.2 System Configuration

Figure 3.3 shows the high-speed robot system. A human throws a ball toward the manipulator from 2.5 m distance. The manipulator hits the ball toward an objective point. Figure 3.4(b) shows a photo of the system.

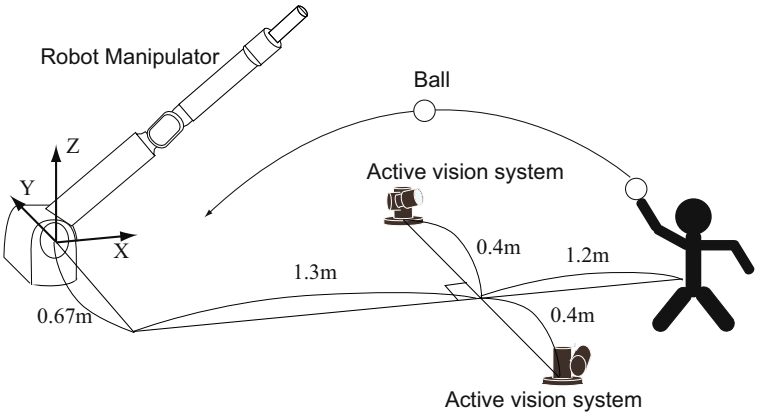
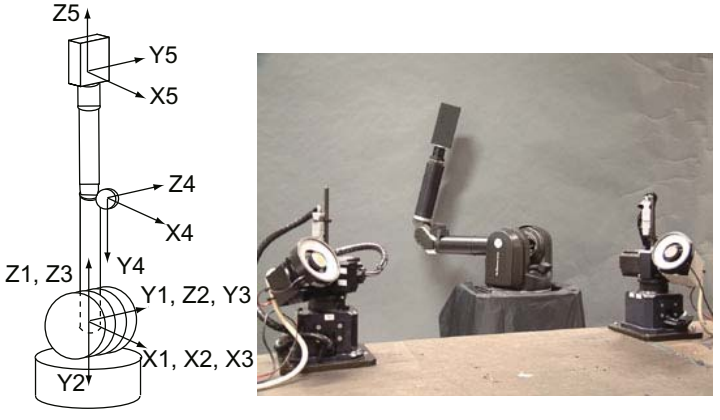


Fig. 3.3 System configuration [10]



(a) Kinematics (b) Manipulator and vision system

Fig. 3.4 System components [10]

3.2.2.1 Manipulator

The kinetic system consists of a wire-drive manipulator made in Barrett Technology Inc. [8]. The kinematics of the manipulator is shown in Figure 3.4(a). The manipulator has 5 degrees of freedom (DOF) joints, and a plane plate is set as a bat in order to control the direction of the hit ball. The maximum velocity of the end-effector is about 8 m/s, and its maximum acceleration is about 58 m/s^2.

3.2.2.2 Active Vision

The visual processing is executed in a massive parallel vision system called column-parallel high-speed vision system (CPV) [6]. It consists of a 128×128 image sensor

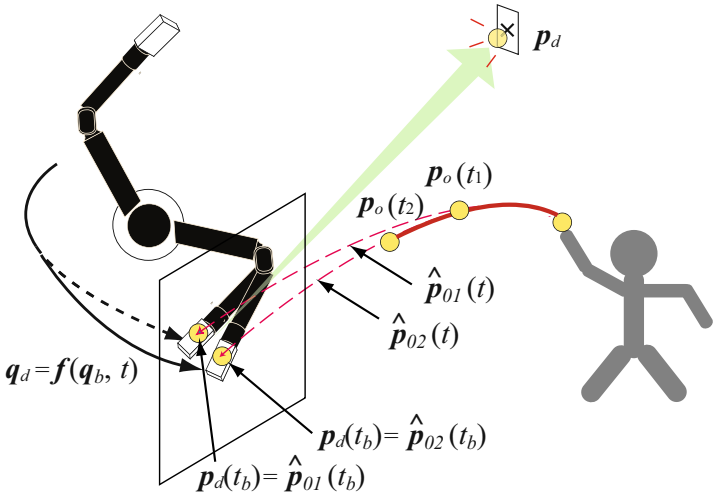


Fig. 3.5 Ball control in batting motion [10]

and a 128×128 parallel processing array. The image sensor is connected with the parallel processing array by a special parallel signal cable. Because the visual processing is executed in parallel in the processing array, high-speed visual processing (moment detection, segmentation, *etc*) is achieved within 1 ms.

Each vision sensor is mounted on an active vision, which has 2 DOF (the pan and tilt). By integrating the image information with the joint angle information, the 3D position is calculated precisely. The information acquired by the vision system is sent to the manipulator at the rate of 1 kHz.

3.2.3 Batting Algorithm

The desired trajectory of the manipulator is generated by the following three steps:

1. measure the position of the target $\mathbf{p}_o$ by stereo vision, and estimate the position of the target at the hitting time $t = t_b$;
2. compute the position $\mathbf{p}(t_b)$ and orientation $\phi(t_b)$ of the end-effector at the hitting time $t = t_b$;
3. generate the desired trajectory $\mathbf{q}_d$ satisfying the boundary condition.

In every cycle time, the steps 1 and 2 are repeated from the first time the vision recognizes the target until the time the manipulator swings ($t = 0$), and the steps 1, 2 and 3 are repeated until the hitting time ($t = t_b$). This means that the hitting point is estimated before impact explicitly, and the batting task is accomplished by modifying this point based on visual feedback. The detailed processes are explained in the following.

Step 1. First, the 3D visual information is calculated by means of the stereo active vision. By image-based visual servoing, the active vision executes 3D tracking

so as to keep the target within the field of view. The position of target $\mathbf{p}_0(t)$ is computed using the center of gravity of the acquired image. Next, the future trajectory of the target object is estimated as a polynomial function of the time t such as $\hat{\mathbf{p}}_{0n}(t)$. The estimated trajectory $\hat{\mathbf{p}}_{0n}(t)$ is fitted in with the past sequence of the target positions such as $\mathbf{p}_0(t_n), \mathbf{p}_0(t_{n-1}), \cdots, \mathbf{p}_0(t_1)$ based on successive least-squares estimation, where t_n is the sampling time, and n is the number of sampling.

Step 2. First, we set a virtual plane expressed as

$$ax + by + cz = d, \tag{3.2}$$

where a, b, c, and d are constants. The plane is defined so that the ball hits the bat in it. By solving the equation

$$\begin{bmatrix} a & b & c \end{bmatrix} \hat{\mathbf{p}}_{0n}(t) = d \tag{3.3}$$

we get the hitting time $t = t_b$. As a result, the desired position $\mathbf{p}_d(t_b) = \hat{\mathbf{p}}_{0n}(t_b)$ of the end-effector at the hitting time is computed. In general, $t_n < t_b$, and n should be a large number to achieve accurate estimation. Then, the desired orientation $\phi_d(t_b)$ at the hitting time is arbitrarily given in order to control the direction of the hit ball.

Step 3. The joint angle vector at the hitting point is $\mathbf{q}_b \in \mathbb{R}^n$, and the boundary condition is written as

$$\mathbf{q}_b = \mathbf{q}_d(t_b) = \mathbf{l}^{-1}(\mathbf{p}_d(t_b), \phi_d(t_b)) \tag{3.4}$$

where the function $\mathbf{l}^{-1}(\dot{})$ means the inverse kinematics.

In order to use the past sequence of the ball position with the current ball position, we modify (3.1) to

$$\mathbf{q}_d = \mathbf{f}(\mathbf{q}_b, t). \tag{3.5}$$

We adopt a fifth order polynomial as the trajectory function $\mathbf{f}$, in order to control the position, the velocity, and the acceleration, continuously:

$$\mathbf{q}_d(t) = \sum_{i=0}^{5} \mathbf{k}_i(\mathbf{q}_b)\, t^i \ . \tag{3.6}$$

As a result, the trajectory of a manipulator is determined by the coefficients $\mathbf{k}_i$. The coefficients $\mathbf{k}_i$ are

$$\mathbf{k}_0 = \mathbf{q}_d(0) \tag{3.7}$$

$$\mathbf{k}_1 = \dot{\mathbf{q}}_d(0) \tag{3.8}$$

$$\mathbf{k}_2 = \frac{1}{2}\ddot{\mathbf{q}}_d(0) \tag{3.9}$$

$$\mathbf{k}_3 = \frac{1}{2t_b^3}\left\{20\left[\mathbf{q}_b - \mathbf{q}_d(0)\right] - t_b\left[8\,\mathbf{c}_v + 12\dot{\mathbf{q}}_d(0)\right]x + t_b^2\left[\mathbf{c}_a - 3\ddot{\mathbf{q}}_d(0)\right]\right\} \tag{3.10}$$

$$\mathbf{k}_4 = \frac{1}{2t_b^4}\left\{-30\left[\mathbf{q}_b-\mathbf{q}_d(0)\right]+t_b\left[14\,\mathbf{c}_v+16\dot{\mathbf{q}}_d(0)\right]-t_b^2\left[2\mathbf{c}_a-3\ddot{\mathbf{q}}_d(0)\right]\right\} \quad (3.11)$$

$$\mathbf{k}_5 = \frac{1}{2t_b^5}\left\{12\left[\mathbf{q}_b-\mathbf{q}_d(0)\right]-6t_b\left[\mathbf{c}_v+\dot{\mathbf{q}}_d(0)\right]+t_b^2\left[\mathbf{c}_a-\ddot{\mathbf{q}}_d(0)\right]\right\} \quad (3.12)$$

where $\mathbf{c}_v$, $\mathbf{c}_a$ represent arbitrary vectors. If the hitting time and swing time are constant during the swing motion, the vectors are represented by $\mathbf{c}_v=\dot{\mathbf{q}}_d(t_b)$, $\mathbf{c}_a=\ddot{\mathbf{q}}_d(t_b)$ respectively. Thus we can set the speed of the manipulator at the hitting point by adjusting the arbitrary vectors.

To increase the velocity of the end-effector in the serial link mechanism, it is necessary to speed up the velocity of the joints on the bottom side. Then it is easy to modify the trajectory of the joints on the top side during the swing motion, so that the manipulator can track an unpredictable target motion. This is because the inertia of the top side is lower than that of the bottom side. For this reason, the bottom side is controlled only in the SW mode, and the top side is controlled in the HT mode.

Define the vector $\mathbf{q}_b^{sw}$ as

$$\mathbf{q}_b^{sw} = \mathbf{l}^{-1}\left(\hat{\mathbf{p}}_{01}(t_b), \phi_d(t_b)\right). \quad (3.13)$$

It is the desired joint angle at the hitting time in the case that the target trajectory is estimated only by using one sample $\mathbf{p}_0(t_1)$, and it means that the desired trajectory of the manipulator is not modified during the swing motion. Next, define the vector $\mathbf{q}_b^{int} = [q_{b1}^{sw}, q_{b2}^{sw}, q_{b3}^{top}, q_{b4}^{top}, q_{b5}^{top}]^t$, where q_{b1}^{sw} and q_{b2}^{sw} are respectively the first and second element of the vector $\mathbf{q}_b^{sw}$. The parameters q_{b3}^{top}, q_{b4}^{top}, q_{b5}^{top} are decided so that the following equation holds:

$$\mathbf{q}_b^{int} = \mathbf{l}^{-1}\left(\mathbf{p}_d(t_b), \phi_d(t_b)\right). \quad (3.14)$$

By using $\mathbf{q}_b^{int}$ instead of $\mathbf{q}_b$ in (3.6), the joints 1 and 2 are controlled in the SW mode, and the joints 3, 4, and 5 are controlled in the integration of the SW and HT mode.

3.2.4 Experiment

In the experiment, a human threw a styrofoam ball with radius 5 cm toward the manipulator from 2.5 m distance, and the manipulator hits the ball toward the objective point $\mathbf{p}_d = [1.9\ 0.0\ 1.3]^T$, and $\dot{\mathbf{q}}(0) = \ddot{\mathbf{q}}(0) = \mathbf{0}$, the hitting time $t_b = 0.25$ s, and the whole swing time is 0.85 s. In addition, we use a proportional derivative (PD) controller to track the desired trajectory of the manipulator, and the second joint q_2 is fixed during the swing motion for safety. After the hitting, the trajectory of the manipulator is also generated by another fifth order polynomial for the follow-through. This allows the manipulator to stop smoothly.

From the moment the vision system finds the ball, it takes the ball $0.3 \sim 0.4$ s to reach the hitting point. In this experiment, the manipulator must immediately start to swing just after the vision sensor recognizes the ball, and its period is only 0.1 s.

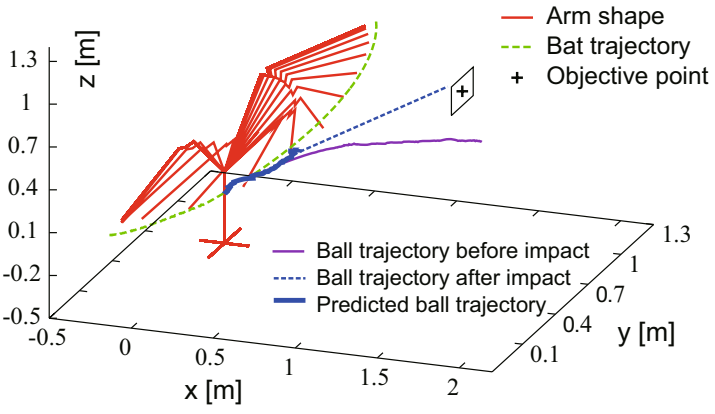
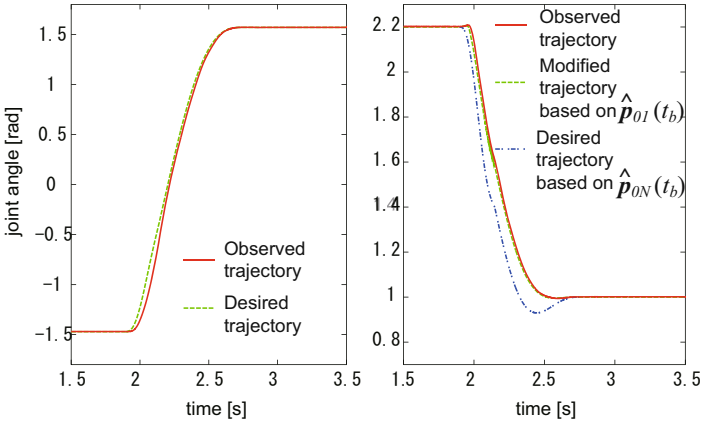


Fig. 3.6 Batting motion [10]



(a) SW mode (joint 1) (b) HT mode (joint 4)

Fig. 3.7 Time response of joint angles [10]

The velocity of the ball is $6 \sim 8$ m/s, and the velocity of the end-effector is about 6 m/s at the impact point.

Figure 3.6 shows the motion of the arm and ball. The ball is recognized at $x = 2.1$ m and is hit on the hitting point at $x = 0.33$ m. From the data for the ball position after hitting, it turns out that the hit ball heads in the direction to the objective point. The time response of joint angles is shown in Figure 3.7. It turns out that the smooth joint trajectory is generated in either mode. In HT mode (the joint 4), the desired trajectory based on $\hat{\mathbf{p}}_{01}(t_b)$ is modified to that based on $\hat{\mathbf{p}}_{0N}(t_b)$ $(N = 250)$ due to the shift of the hitting point. Then the actual trajectory of the manipulator tracks it. This result means that the manipulator can hit a breaking ball.

Fig. 3.8 Continuous sequence of batting motion [10]

In this way, even though a simple PD controller is used, the manipulator achieves high-speed and reactive motion by giving the appropriate desired command. Figures 3.8 and 3.9 are continuous sequences of pictures which were taken at the intervals of 132 ms and 30 ms respectively.

Figure 3.10 shows the simulation results of the time response of the desired joint velocity and the desired motor torque when the hitting point moves 10 cm during the swing motion. The desired joint velocity ω_d and the desired motor torque τ_d are computed as

$$\omega_d = \dot{\mathbf{q}}_d, \tag{3.15}$$

$$\tau_d = \mathbf{M}(\mathbf{q}_d)\ddot{\mathbf{q}}_\mathbf{d} + \mathbf{h}(\mathbf{q}_d, \dot{\mathbf{q}}_d) + \mathbf{g}(\mathbf{q}_d). \tag{3.16}$$

Even in this case, a smooth trajectory is generated which is similar to the trajectory in the case of the fixed hitting point. The change of torque is continuous, but not smooth at the start time and the hitting time. This is because the polynomial

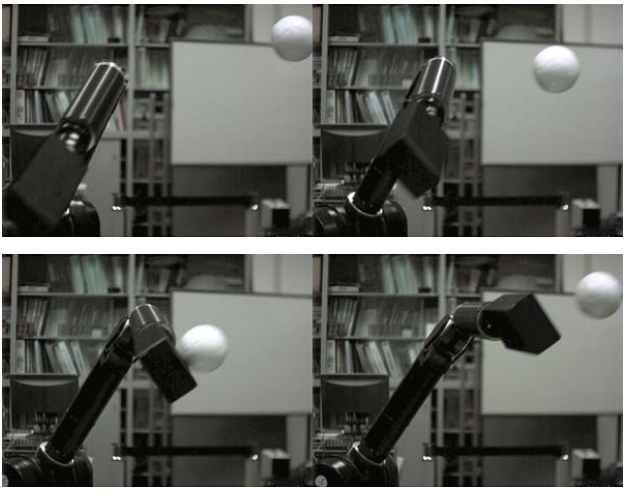
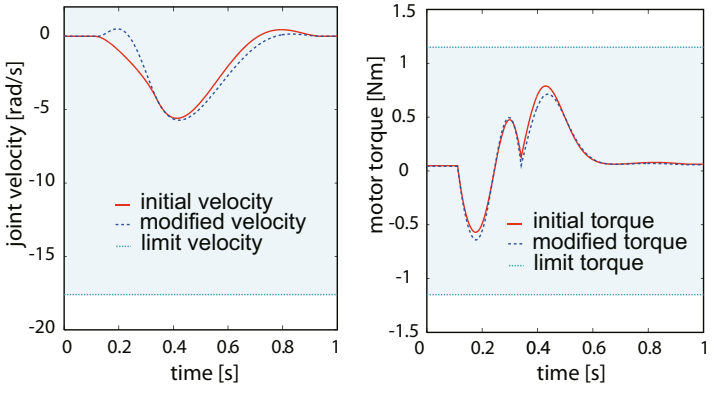


Fig. 3.9 Continuous sequence of batting motion (close view) [10]



(a) Velocity response (b) Torque response

Fig. 3.10 Time response of velocity and torque [10]

trajectory is switched to the other polynomial trajectory of the follow-through at the hitting time. This problem is solved by generating a function with consideration to the jerk.

3.3 Tool Manipulation by Visual Servoing

One of the dexterous tasks of a human hand is tool manipulation. A human hand can operate many types of tools. The tool manipulation is more difficult than the typical

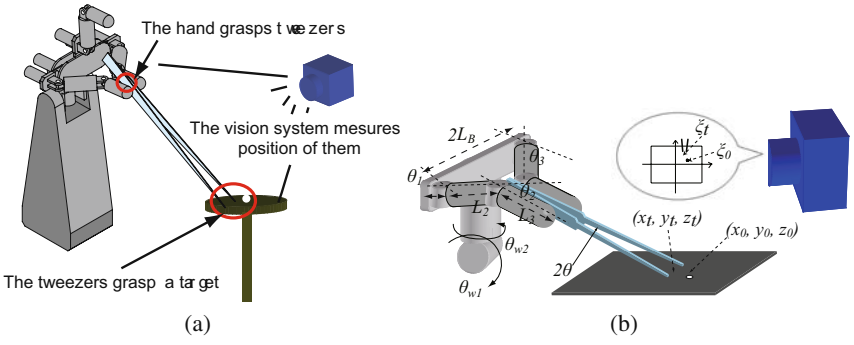


Fig. 3.11 (a) System configuration, and (b) visual servo control [5]

manipulation. The object also has to be controlled by the other grasped object. In order to achieve such a task, visual feedback control is very important [5].

3.3.1 Passive Joint in the Contact Point

As a handled tool, we adopt tweezers. Figure 3.11(a) shows the setup of the experiment. A multifingered hand grasps tweezers with two fingers. The hand manipulates the tip of the tweezers so that the tweezers pick up the object placed on the table. The position of hand, tweezers and object are measured by vision system. In this configuration, the hand and tweezers can be regarded as one mechanism, and the contact points between the fingers and tweezers are regarded as a kind of passive joint.

It is possible that the condition at the passive joints may change during manipulation. First, the initial grasp condition of the tweezers may be changed at each trial. Next, if some external force is added to a part of the tweezers during manipulation, the rotational moment at the passive joints occurs. When the rotational moments are larger than the static frictional force, the relative posture of the tweezers to the hand is also changed. For these reasons, it is necessary to measure the relative positions in realtime and to control the hand so as to cancel the changes. Visual servoing control [2] is effective for this problem. If the rate of feedback is sufficiently fast, the control can be achieved more smoothly.

3.3.2 Visual Servo Control for Tool Manipulation

The hand is controlled by image based visual servoing shown in Figure 3.11(b). Image based visual servoing is a control method by feedback of the position error in the image plane. It is robust against the calibration error, and the computation of the target 3D position is not needed. The fingers are used to control the tweezers grip, and the 2-axis wrist joints $\theta_w = [\theta_{w1}, \theta_{w2}]^T$ are used to control the position of the tip of the tweezers.

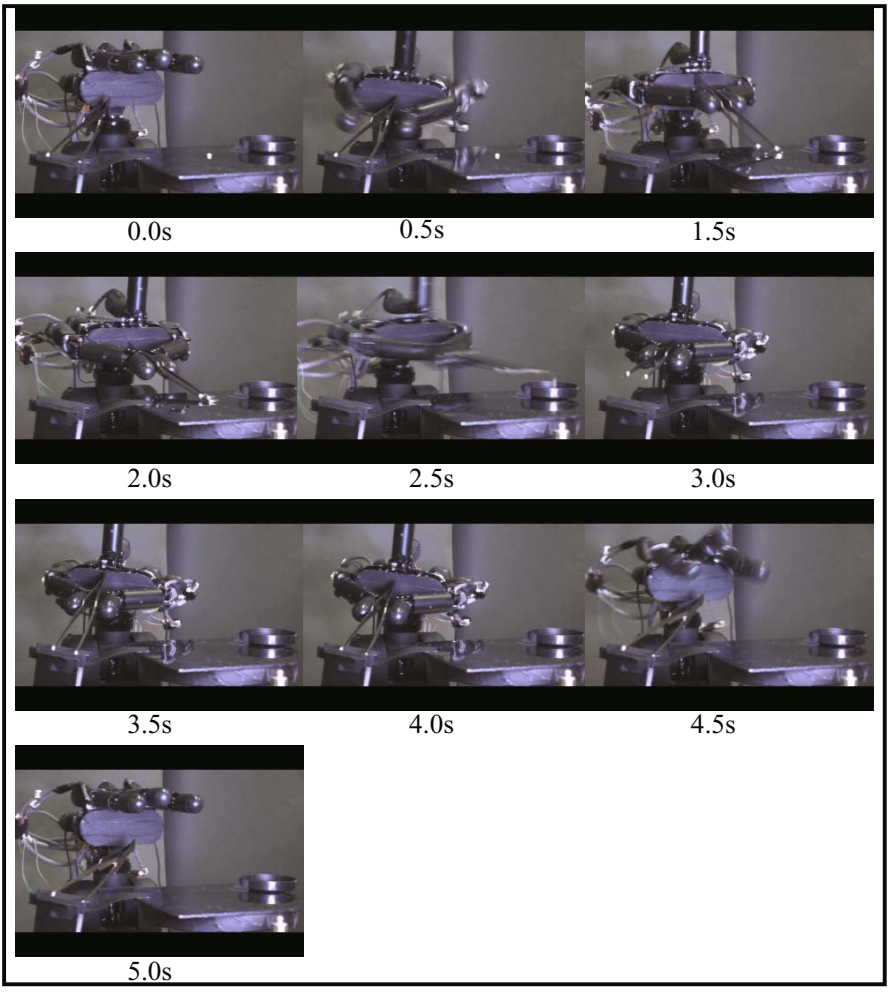


Fig. 3.12 Continuous photograph of tool manipulation [5]

The target position in the image plane is defined as $\xi = [\xi_x, \xi_y]^T$ and the 3D position of the target is defined as $\mathbf{x} = [x, y, z]^T$. Using the perspective projection function $\mathbf{f}$, we can write $\xi = \mathbf{f}(\mathbf{x})$. From the kinematics of the hand and tweezers structure, we can write the tip of tweezers point as $\mathbf{x} = \mathbf{g}(\theta)$. By differentiating this relation we get

$$\dot{\xi} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \frac{\partial \mathbf{g}}{\partial \theta_w} \dot{\theta}_w = \mathbf{J} \dot{\theta}_w \tag{3.17}$$

where $\mathbf{J}$ is an image Jacobian. Using this relationship, the desired wrist joint angle θ_{dw} is computed as

$$\theta_{dw} = k\mathbf{J}^{-1}(\xi_d - \xi) + \theta_w, \tag{3.18}$$

where ξ_d is the coordinates of the target in the image plane, ξ is the coordinate of the tip of the tweezers in the image plane, and k is constant.

Using this equation, we can calculate the desired angle of the wrist joint. When the tip of the tweezers touches the table plane where the target is placed, the impact is absorbed by the passive joint, and the tip of the tweezers can move along the surface of the table plane. In this condition, the image Jacobian $\mathbf{J}$ is changed because of the change in kinematics of the grip. Although the speed of convergence is changed, tracking is achieved in most cases because of robustness of the visual servoing.

In order to improve the speed of convergence, the feedback gain is changed as

$$k \leftarrow k + a\left(b - \left|\frac{\partial \xi_t}{\partial \mathbf{x}}\right|\right), \tag{3.19}$$

where $\frac{\partial \xi_t}{\partial \mathbf{x}}$ is the rate of change of image feature, and a, b are constant coefficients. It corresponds to the tracking speed of the tweezers, and the gain is automatically adjusted.

3.3.3 Experiments

The multifingered hand has three fingers [7]. The middle finger has 2 DOF, and the other fingers have 3 DOF, and the hand has 8 DOF in total. Each finger can close at 180 deg per 0.1 s. The hand is mounted on the 2-axis wrist mechanism. The vision system uses a high-speed camera Basler A504k. It takes 8 bit gray scale images and the maximum speed is 500 fps by 1280×1024 resolution.

We used a single rice grain as a target; its long axis is about 5 mm and its short axis is about 3 mm. Figure 3.12 shows the continuous photographs of total processes. Figure 3.13 shows the part of the target tracking of tweezers to the target.

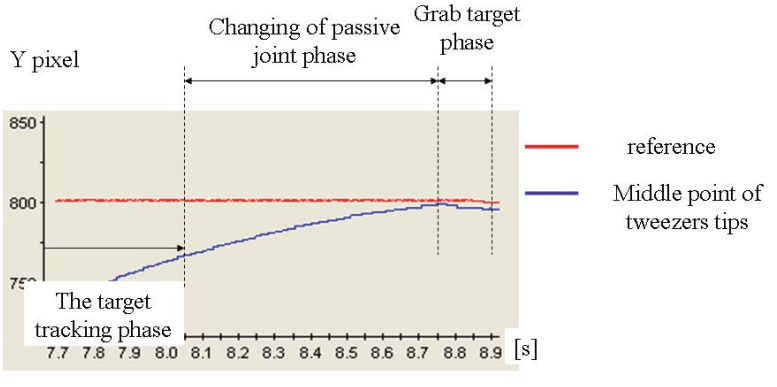


Fig. 3.13 Time response [5]

When Y pixel of the center of tweezers was 745, the tip of tweezers touched the table plane. After that, the tweezers tips moved along the surface. Normally, if the passive joint structure changes, the tracking speed changes rapidly. In this case, the speed changed smoothly by changing the coefficient of the control method.

3.4 Conclusion

In this chapter, we described some examples of visual feedback control methods in our high-speed manipulation system. First, a hybrid trajectory generator using visual information is proposed for a batting task. Next, visual servoing control including passive joint problem is proposed for tool manipulation. Visual information is one of the most important information for manipulation. However, the real-time visual feedback control is not enough applied to manipulation system. The high-speed visual feedback control is one of the breakthrough methods. In the future, it will improve the ability of manipulation systems.

References

[1] Furukawa, N., Namiki, A., Senoo, T., Ishikawa, M.: Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 181–187 (2006)
[2] Hutchinson, S., Grefory, G., Corke, P.: A tutorial on visual servo control. IEEE Trans. on Robotics and. Automation 12(5), 651–670 (1996)
[3] Ishihara, T., Namiki, A., Ishikawa, M., Shimojo, M.: Dynamic pen spinning using a high-speed multifingered hand with high-speed tactile sensor. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 258–263 (2006)
[4] Ishikawa, M., Kazuya Ogawa, T.K., Ishii, I.: A cmos vision chip with simd processing element array for 1ms image processing. In: Dig. Tech. Papers of IEEE Int. Solid-State Circuits Conf., pp. 206–207 (1999)
[5] Mizusawa, S., Namiki, A., Ishikawa, M.: Tweezers type tool manipulation by a multi-fingered hand using a high-speed visual servoing. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2709–2714 (2008)
[6] Nakabo, Y., Ishikawa, M., Toyoda, H., Mizuno, S.: 1ms column parallel vision system and its application of high speed target tracking. In: IEEE Int. Conf. Robotics and Automation, pp. 650–655 (2000)
[7] Namiki, A., Imai, Y., Ishikawa, M., Kaneko, M.: Development of a high-speed multi-fingered hand system and its application to catching. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 2666–2671 (2003)
[8] Rooks, B.: The harmonious robot. Industrial Robot: An International Journal 33(2), 125–130 (2006)
[9] Senoo, T., Namiki, A., Ishikawa, M.: High-speed batting using a multi-jointed manipulator. In: IEEE Int. Conf. Robotics and Automation, pp. 191–1196 (2004)

[10] Senoo, T., Namiki, A., Ishikawa, M.: Ball control in high-speed batting motion using hybrid trajectory generator. In: IEEE Int. Conf. Robotics and Automation, pp. 1762–1767 (2006)

[11] Yamakawa, Y., Namiki, A., Ishikawa, M., Shimojo, M.: One-handed knotting of a flexible rope with a high-speed multifingered hand having tactile sensors. In: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 703–708 (2007)

Chapter 4
Human-machine Cooperative Manipulation with Vision-based Motion Constraints

Gregory D. Hager

Abstract. This chapter discusses a class of control algorithms that provide enhanced physical dexterity by imposing passive motion constraints. Such motion constraints are often referred to as virtual fixtures. It is shown that algorithms originally designed for vision-based control of manipulators can be easily converted into control algorithms that provide virtual fixtures. As a result it is possible to create advanced human-machine cooperative manipulation systems that take complete advantage of information provided by vision, yet permit the user to retain control of essential aspects of a given task.

4.1 Introduction

Much of "classical" robotics has focused on creating machines that are autonomous. However, such endeavors are fundamentally limited by our ability to create machines that can perceive, judge, and react to unforeseen (or sometimes foreseen!) circumstances in the world. To this day, there are still few situations, other than rote, open-loop manipulation, where robotics have even begun to compete with humans, in this regard.

At the other extreme, teleoperation tends to focus on providing a high-performance, high-fidelity operator interface to a machine. In many ways, this is a natural marriage, as the human now controls a machine that may be more accurate, reliable, or powerful than the human operator. However, while the machine is clearly amplifying some level of human skill, it does so at the cost of attenuating others. In particular, teleoperation systems do not have an underlying representation of the *intent* of the operator. As a result, it is not possible to adapt or modify the interface to enhance those skills that are most germaine to the task at hand.

Our group in the center for computer integrated surgical systems (CISST) has been working to create surgical systems that improve both the speed and precision

Gregory D. Hager
Department of Computer Science, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA
e-mail: hager@cs.jhu.edu

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 55–70.
springerlink.com

of medical interventions. Our goal is to create mechanisms that are neither autonomous, nor purely passive. Rather, our intent is to create mechanisms that selectively provide cooperative assistance to a surgeon, while allowing the surgeon to retain ultimate control of the procedure.

In our recent work, we have focused on developing assistance methods for microsurgery. Here, the extreme challenges of physical scale accentuate the need for dexterity enhancement, but the unstructured nature of the tasks dictates that the human be directly "in the loop". For example, retinal vein cannulation [31] involves the insertion of a needle of approx. 20-50 microns in diameter into the lumen of a retinal vein (typically 100 microns in diameter or less)[1]. At these scales, tactile feedback is practically non-existent, and depth perception is limited to what can be seen through a stereo surgical microscope. In short, such a procedure is at the limit of what is humanly possible in conventional surgical practice.

Given the scale of operation, the most obvious need is to increase the precision of human motion, ideally without slowing or limiting the surgeon. In recent work [19, 20, 18], we have begun to develop assistant methods that are based on manipulating the apparent compliance of tools simultaneously held by both the surgeon and a robot. Intuitively, if a tool is extremely stiff, then it is easier to achieve high precision of motion, and to remove tremor. Conversely, low stiffness makes it possible to perform large-scale "transport" motions.

Although they increase absolute precision, purely isotropic compliances cannot take advantage of natural task constraints to provide structured assistance. For example, when placing a needle into the lumen of a blood vessel, the natural mode of assistance would be to stabilize the needle in the lateral directions, but permit relatively free, quasi-static positioning along the needle axis.

In this chapter, we specifically focus on the use of *anisotropic* compliances as a means of assistance. In previous work we have related these anisotropic compliances to the notion of virtual fixtures [21, 28]. Virtual fixtures, like the real thing, provide a surface that confines and/or guides motion. We initially describe how virtual fixtures can be produced as a generalization of previous work in [1, 2, 24], and then turn to the problem of deriving vision-based virtual fixtures. Through example, we show how visual servoing algorithms for one camera [3] and two camera [8, 10] systems can be translated into virtual fixtures. As a result, much of the previous literature on visual servoing can be applied to the problem of human-machine cooperative manipulation. Finally, we describe several applications, both surgical and nonsurgical, that we have developed around virtual fixtures of the type introduced in this chapter.

4.2 Virtual Fixtures

Our work has been motivated by the Johns Hopkins University (JHU) steady hand robot (SHR), and, in particular, the assistance paradigm of direct manipulation it was designed for [18, 19, 30]. Briefly, the JHU SHR is a 7 degrees of freedom (DOF)

[1] As a point of reference, a human hair is typically on the order of 80 microns in diameter.

robot equipped with a force sensing handle at the endpoint. Tools are mounted at the endpoint, and "manipulated" by an operator holding the force handle. The robot responds to the applied force, thus implementing a means of direct control for the operator. The robot has been designed to provide micron-scale accuracy, and to be ergonomically appropriate for minimally invasive microsurgical tasks [30].

In this section, we introduce the basic admittance control model used for the SHR, extend this control to anisotropic compliances, and finally relate anisotropic compliances to an underlying task geometry.

In the remainder of this chapter, transpose is denoted by $'$, scalars are written lowercase in normal face, vectors are lowercase and boldface, and matrices are normal face uppercase.

4.2.1 Virtual Fixtures as a Control Law

In what follows, we model the robot as a purely kinematic Cartesian device with tool tip position $\mathbf{x} \in SE(3)$ and a control input that is endpoint velocity $\mathbf{v} = \dot{\mathbf{x}} \in \Re^6$, all expressed in the robot base frame. The robot is guided by applying forces and torques $\mathbf{f} \in \Re^6$ on the manipulator handle, likewise expressed in robot base coordinates.

In the steady-hand paradigm, the relationship between velocity and motion is derived by considering a "virtual contact" between the robot tool tip and the environment. In most cases, this contact is modeled by a linear viscous friction law

$$k\mathbf{v} = \mathbf{f}, \tag{4.1}$$

or equivalently

$$\mathbf{v} = \frac{1}{k}\mathbf{f}, \tag{4.2}$$

where $k > 0$ controls the stiffness of the contact. In what follows, it will be more convenient to talk in terms of a compliance $c \equiv 1/k$.

When using (4.2), the effect is that the manipulator is equally compliant in all directions. Suppose we now replace the single constant c with a diagonal matrix C. Making use of C in (4.2) gives us the freedom to change the compliance of the manipulator in the coordinate directions. For example, setting all but the first two diagonal entries to zero would create a system that permitted motion only in the x-y plane. It is this type of anisotropic compliance that we term a virtual fixture. In the case above, the fixture is "hard " meaning it permits motion in a subspace of the workspace. If we instead set the first two entries to a large value, and the remaining entries to a small one, the fixture becomes "soft." Now, motion in all directions is allowed, but some directions are easier to move in than others. We refer to the motions with high compliance as *preferred* directions, and the remaining directions as *nonpreferred* directions.

4.2.2 Virtual Fixtures as Geometric Constraints

While it is clearly possible to continue to extend the notion of virtual fixture purely in terms of compliances, we instead prefer to take a more geometric approach, as suggested in [1, 2]. We will develop this geometry by specifically identifying the preferred and non-preferred directions of motion at a given time point t. To this end, let us assume that we are given a $6 \times n$ time-varying matrix $D = D(t)$, $0 < n < 6$. Intuitively, D represents the instantaneous preferred directions of motion. For example, if n is 1, the preferred direction is along a curve in $SE(3)$; if n is 2 the preferred directions span a surface; and so forth.

From D, we define two projection operators, the span and the kernel of the column space, as

$$\text{Span}(D) \equiv [D] = D(D'D)^{-1}D' \tag{4.3}$$

$$\text{Ker}(D) \equiv \langle D \rangle = I - [D]. \tag{4.4}$$

This formulation assumes that D has full column rank. It will occasionally be useful to deal with cases where the rank of D is lower than the number of columns (in particular, the case when $D = 0$). For this reason, we will assume $[\,\cdot\,]$ has been implemented using the pseudo-inverse [29, pp. 142–144] and write

$$\text{Span}(D) \equiv [D] = D(D'D)^{+}D' \tag{4.5}$$

$$\text{Ker}(D) \equiv \langle D \rangle = I - [D]. \tag{4.6}$$

The following properties hold for these operators [29]:

1. symmetry: $[D] = [D]'$;
2. idempotence: $[D] = [D][D]$;
3. scale invariance: $[D] = [kD]$;
4. orthogonality: $\langle D \rangle' [D] = 0$;
5. completeness: $\text{rank}(\alpha \langle D \rangle + \beta [D]) = n$ where D is $n \times m$ and $\alpha, \beta \neq 0$;
6. equivalence of projection: $[\langle D \rangle f] f = \langle D \rangle f$.

The above statements remain true if we exchange $\langle D \rangle$ and $[D]$. Finally, it is useful to note the following equivalences:

- $[[D]] = [D]$;
- $\langle \langle D \rangle \rangle = [D]$; and
- $[\langle D \rangle] = \langle [D] \rangle = \langle D \rangle$.

Returning to our original problem, consider now decomposing the input force vector, $\mathbf{f}$, into two components

$$\mathbf{f}_D \equiv [D]\mathbf{f} \quad \text{and} \quad \mathbf{f}_\tau \equiv \mathbf{f} - \mathbf{f}_D = \langle D \rangle \mathbf{f}. \tag{4.7}$$

It follows directly from property 4 that $\mathbf{f}_D \cdot \mathbf{f}_\tau = 0$ and from property 5 that $\mathbf{f}_D + \mathbf{f}_\tau = \mathbf{f}$. Combining (4.7) and (4.2), we can now write

$$\mathbf{v} = c\mathbf{f} = c(\mathbf{f}_D + \mathbf{f}_\tau). \tag{4.8}$$

Let us now introduce a new compliance $c_\tau \in [0,1]$ that attenuates the non-preferred component of the force input. With this we arrive at

$$\begin{aligned}\mathbf{v} &= c(\mathbf{f}_D + c_\tau \mathbf{f}_\tau) \\ &= c([D] + c_\tau \langle D \rangle)\mathbf{f}.\end{aligned} \tag{4.9}$$

Thus, the final control law is in the general form of an admittance control with a time-varying gain matrix determined by $D(t)$. By choosing c, we control the overall compliance of the system. Choosing c_τ low imposes the additional constraint that the robot is stiffer in the non-preferred directions of motion. As noted above, we refer to the case of $c_\tau = 0$ as a *hard virtual fixture*, since it is not possible to move in any direction other than the preferred direction. All other cases will be referred to as *soft virtual fixtures.* In the case $c_\tau = 1$, we have an isotropic compliance as before.

It is also possible to choose $c_\tau > 1$ and create a virtual fixture where it is easier to move in non-preferred directions than preferred. In this case, the natural approach would be to switch the role of the preferred and non-preferred directions.

4.2.3 Choosing the Preferred Direction

The development to this point directly supports the following types of guidance:

- motion in a subspace: suppose we are supplied with a time-varying, continuous function $D = D(t)$. Then applying (4.9) yields a motion constraint within that subspace; and
- motion to a target pose $\mathbf{x}_t \in SE(3)$: suppose that we have a control law $\mathbf{u} = f(\mathbf{x}, \mathbf{x}_t)$ such that by setting $\mathbf{v} = \mathbf{u}$,

$$\lim_{t \to \infty} \mathbf{x} = \mathbf{x}_t.$$

  Then by choosing $D = \mathbf{u}$ and applying (4.9), we create a virtual fixture that guides the user to the given target pose.

These two tasks are, in some sense, at the extremes of guidance. In one case, there is no specific objective to attain; we are merely constraining motion. In the second, the pose of the manipulator is completely constrained by the objective. What of tasks that fall between these two extremes?

To study this problem, let us take a simple yet illustrative case: the case of maintaining the tool tip within a plane through the origin. For the moment, let us neglect manipulator orientation and consider the problem when controlling just the spatial position of the endpoint. We define the surface as $P(\mathbf{p}) = \mathbf{n} \cdot \mathbf{p} = 0$ where $\mathbf{n}$ is a unit vector expressed in robot base coordinates.

Based on our previous observations, if the goal was to allow motion *parallel* to this plane, then, noting that $\mathbf{n}$ is a non-preferred direction in this case, we would define $D = \langle \mathbf{n} \rangle$ and apply (4.9). However, if the tool tip is not in the plane, then it is

necessary to adjust the preferred direction to move the tool tip toward it. Noting that $P(\mathbf{x})$ is the (signed) distance from the plane, we define a new preferred direction as follows:

$$D_c(\mathbf{x}) = [(1-k_d)\langle \mathbf{n} \rangle \mathbf{f}/\|\mathbf{f}\| - k_d[\mathbf{n}]\mathbf{x}] \quad 0 < k_d < 1. \tag{4.10}$$

The geometry of (4.10) is as follows. The idea is to first produce the projection of the applied force onto the nominal set of preferred directions, in this case $\langle \mathbf{n} \rangle$. At the same time, the location of the tool tip is projected onto the plane normal vector. The convex combination of the two vectors yields a resultant vector that will return the tool tip to the plane. Choosing the constant k_d governs how quickly the tool is moved toward the plane. One minor issue here is that the division by $\|\mathbf{f}\|$ is undefined when no user force is present. Anticipating the use of projection operators (which are scale invariant, as noted earlier), we make use of a scaled version of (4.10) that does not suffer this problem:

$$D_c(\mathbf{x}) = (1-k_d)\langle \mathbf{n} \rangle \mathbf{f} - k_d\|\mathbf{f}\|[\mathbf{n}]\mathbf{x} \quad 0 < k_d < 1. \tag{4.11}$$

We now apply (4.9) with $D = D_c$. Noting that the second term on the right hand side could also be written

$$\|\mathbf{f}\| k_d P(\mathbf{x})\mathbf{n},$$

it is easy to see that, when the tool tip lies in the plane, the second term vanishes. In this case, it is not hard to show, using the properties of the projection operators, that combining (4.11) with (4.9) results in a law equivalent to a pure subspace motion constraint. One potential disadvantage of this law is that when user applied force is zero, there is no virtual fixture as there is no defined preferred direction. Thus, there is a discontinuity at the origin. However, in practice the resolution of any force sensing device is usually well below the numerical resolution of the underlying computational hardware, so the user will never experience this discontinuity.

With this example in place, it is not hard to see its generalization to a broader set of control laws. We first note that another way of expressing this example would be to posit a control law of the form

$$\mathbf{u} = -(\mathbf{n} \cdot \mathbf{x})\mathbf{n} = -[\mathbf{n}]\mathbf{x}, \tag{4.12}$$

and to note that assigning $\mathbf{v} = \mathbf{u}$ would drive the manipulator into the plane. This is, of course, exactly what appears in the second term of (4.11). If we now generalize this idea, we can state the following informal rule.

General virtual fixture rule. Given:

1. a surface $S \subseteq SE(3)$ (the motion objective) ;
2. a control law $\mathbf{u} = f(\mathbf{x}, S)$ where by setting $\mathbf{v} = \mathbf{u}$,

$$\lim_{t \to \infty} \mathbf{x} \in S$$

(the control law moves the tool tip into S) under a nominal plant model ;

3. a rule for computing preferred directions $D = D(t)$ relative to S where $\langle D \rangle \mathbf{u} = 0$ if and only if $\mathbf{u} = 0$ (the motion direction is consistent with the control law),

then applying the following choice of preferred direction:

$$D_g(\mathbf{x}) = (1 - k_d)[D]\mathbf{f} - k_d \|\mathbf{f}\| \langle D \rangle \mathbf{u} \quad 0 < k_d < 1 \tag{4.13}$$

yields a virtual fixture that controls the robot toward S and seeks to maintain user motion within that surface.

Note that a sufficient condition for condition 3 above to be true is that, for all pairs $\mathbf{u} = \mathbf{u}(t)$ and $D = D(t)$, $[D]\mathbf{u} = 0$. This follows directly from the properties of projection operators given previously.

To provide a concrete example, consider again the problem of moving the tool tip to a plane through the origin, but let us now add the constraint that the tool z axis should be oriented along the plane normal vector. In this case, $\mathbf{n}$ is a preferred direction of motion (it encodes rotations about the z axis that are not important for us). Let $\mathbf{z}$ denote the vector pointing along the tool z axis and define a control law that is

$$\mathbf{u} = \begin{bmatrix} -(\mathbf{x} \cdot \mathbf{n})\mathbf{n} \\ \mathbf{z} \times \mathbf{n} \end{bmatrix}. \tag{4.14}$$

It is easy to see that this law moves the robot into the plane, and also simultaneously orients the end-effector z axis to be along the normal to the plane. Now, let

$$D = D(t) = \begin{bmatrix} \langle \mathbf{n} \rangle & 0 \\ 0 & \mathbf{n} \end{bmatrix}.$$

It follows that $[D]$ is a basis for translation vectors that span the plane, together with rotations about the normal to the plane. Therefore $[D]\mathbf{u} = 0$ since (4.14) produces translations normal to the plane, and rotations about axes that lie in the plane. Thus, the general virtual fixturing rule can be applied.

4.3 Vision-based Virtual Fixtures

Now, we turn to the problem of providing assistance, where the objective defining the virtual fixture is observed by one or more cameras. To simplify the presentation, in what follows we assume that we have calibrated the camera internal parameters and can therefore work in image normalized coordinates [9].

4.3.1 Controlling the Viewer: Pure Translation

Let us start with a well-studied problem. We have a camera fixed to the endpoint of the manipulator, and the camera observes a fixed, static environment. Our goal is to control the motion of the end-effector by defining a motion for the camera itself based on information extracted from the camera image.

To keep things simple, first consider the case of pure translation ($\mathbf{v} \in \Re^3$) where the camera is aligned with the robot base frame. In this case, the relationship between the motion of the camera and the image motion of a fixed point in space is given by the well-known image Jacobian relationship [14]:

$$\dot{\mathbf{h}} = J\mathbf{v}, \tag{4.15}$$

where $\mathbf{h} = (u,v)' \in \Re^2$ is the image location of a feature point, and J is 2×3.

It is again well-known [3, 14] that the rows of J span the (two-dimensional) space of motions that create feature motion in the image, and therefore $\langle J' \rangle$ is the (one-dimensional) space of motions that leave the point fixed in the image. Consider, thus, creating a virtual fixture by defining

$$D = J' \tag{4.16}$$

in (4.9). From the discussion above, it should be clear that this will create a virtual fixture that prefers motion in any direction except along the viewing direction. While it would seem we are done at this point, there is one minor issue: the image Jacobian depends on the depth of the estimated point. However, if we consider the form of the Jacobian in this case, we see it can be written thus:

$$J = \begin{bmatrix} \frac{1}{z} & 0 & \frac{-u}{z} \\ 0 & \frac{1}{z} & \frac{-v}{z} \end{bmatrix} = \frac{1}{z} \begin{bmatrix} 1 & 0 & -u \\ 0 & 1 & -v \end{bmatrix}. \tag{4.17}$$

As such, we see that the term involving z is a scale factor and, as noted earlier, our projection operators are invariant over scaling of their argument. Thus, we have our first result as follows.

Image plane translation. If we restrict $\mathbf{v}$ to be pure translation and choose an image location $\mathbf{h} = (u,v)'$, then implementing (4.9) using

1. $D = J'$ creates a virtual fixture that prefers motion in the plane normal to the viewing direction defined by $\mathbf{h}$ and the camera optical center ;
2. $D = \langle J' \rangle$ creates a virtual fixture that prefers motion along the viewing direction defined by $\mathbf{h}$ and the camera optical center.

As a special case, choosing $u = v = 0$ yields a virtual fixture parallel to the image plane, which, as is obvious from (4.17), is the camera x-y plane.

It is important to note that the image plane virtual fixtures defined above can be implemented both with and without feedback. That is, if we simply choose a fixed image location (*e.g.* the origin), then the camera will always move in a plane orthogonal to the line of sight through the chosen location. On the other hand, if we choose to track a feature over time, then the motion will always be orthogonal to the line of sight to that feature.

The other possibility, with a single feature, is to maintain the visual cue at a specific image location. For example, suppose the goal is to center an observed

point $\mathbf{h}$ in the image. Since our objective is at the origin, we can define a control law of the form

$$\mathbf{u} = -J'\mathbf{h}, \tag{4.18}$$

where J is evaluated at the origin. It is possible to show that this law will converge for any feature starting and remaining in the image [16]. Furthermore, the preferred directions of motion in this case are $\langle J' \rangle$ and so it follows that $\langle \langle J' \rangle \rangle \mathbf{u} = -[J'] J'\mathbf{h} = 0$ only when $\mathbf{h} = 0$. Since $\mathbf{u} \neq 0$ when $\mathbf{h} \neq 0$, we can apply the general virtual fixturing rule. At this point, we can state a useful specialization for the rest of this chapter.

General vision-based virtual fixtures. Suppose we are supplied with an error term $\mathbf{e} = \mathbf{e}(\mathbf{x})$. Let $S = \{\mathbf{x} | \mathbf{e}(\mathbf{x}) = 0\}$, let $J = \partial \mathbf{e} / \partial \mathbf{x}$, and define $\mathbf{u} = WJ'\mathbf{e}$ where W is a symmetric, positive definite matrix of appropriate dimension (*e.g.* $W = (J'J)^+$). Then the general virtual fixture rule can be applied with preferred directions $\langle J' \rangle$ provided $\mathbf{u}$ so computed converges to S under a nominal plant model.

There is an interesting variation on this. Suppose we choose *no* preferred direction of motion (*i.e.* $D = 0$.) In this case, the first term of (4.13) disappears and the preferred direction in (4.9) is simply $\mathbf{u}$. Thus, the result is a virtual fixture that guides the robot to a target position (compare with the rules at the beginning of Section 4.2.3) and then becomes isotropic. Note, however, that by definition $\mathbf{u}$ is always orthogonal to the line of sight, so the camera prefers to maintain a constant distance to the point during motion.

To press home these points, consider a final problem: to place a specific image location on an observed line, and to facilitate motion along the line. Following the development in [10], suppose we observe a fixed line $\mathbf{l} \in \Re^3$, where the three components of $\mathbf{l}$ can be thought of as the normal vector to the line in the image, and the distance from the origin to the line. This vector is also parallel to the normal vector to the plane formed by the optical axis and the line as it appears in the image plane. We also furnish a distinguished image location $\hat{\mathbf{h}} \in \Re^3$, expressed in homogeneous coordinates. We can then define $\mathbf{e} = \hat{\mathbf{h}} \cdot \mathbf{l}$ to be the image-plane distance between the point and the line.

First, we note that the image Jacobian (relative to $\mathbf{e}$) is now simply

$$L = \mathbf{l}' \begin{bmatrix} J \\ 0 \end{bmatrix} \in \Re^3$$

(note that the z dependence in L is once again a non-issue). As we would now expect, L represents *non-preferred* directions of motion, as it spans the space of motions that change the distance from the point to the line. As a result, choosing preferred directions as $\langle L' \rangle$ in (4.9) would prefer camera motion within the plane encoded by $\mathbf{l}$.

In order to actually place the designated point onto the line, we note that the control law

$$\mathbf{u} = L'\mathbf{e} \tag{4.19}$$

will move the feature point $\hat{\mathbf{h}}$ to the observed line [10]. Hence, we apply (4.13) using $D = \langle L' \rangle$ and $\mathbf{u}$ as defined above (in short, another application of the general vision-based virtual fixture rule).

4.3.2 Controlling the Viewer: General Case

In moving from pure translation to general robot motion, almost nothing changes from the previous section other than the increase in dimensionality. In the case of full motion in $SE(3)$, the image Jacobian becomes 2×6 and has the following general form:

$$\begin{bmatrix} \frac{1}{z} & 0 & \frac{-u}{z} & \frac{-uv}{1} & \frac{1^2+u^2}{1} & -v \\ 0 & \frac{1}{z} & \frac{-v}{z} & \frac{-1^2-v^2}{1} & \frac{uv}{1} & u \end{bmatrix}. \tag{4.20}$$

As shown in [12], the kernel of the image Jacobian given in (4.20), is spanned by the four vectors

$$\begin{bmatrix} u \\ v \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ u \\ v \\ 1 \end{bmatrix} \quad \begin{bmatrix} uvz \\ -(u^2+1)z \\ vz \\ -1 \\ 0 \\ u \end{bmatrix} \quad \begin{bmatrix} dz \\ 0 \\ -udz \\ uv \\ -(u^2+1)z \\ u \end{bmatrix} \tag{4.21}$$

where $d = (u^2 + v^2 + 1)$. As such, we can see that the kernel spans motions that include: 1) motion along the line of sight; 2) rotation about the line of sight; 3) motion on a sphere linking the point with the camera; and 4) linear combinations thereof. Note that 3) spans 2 DOF.

If we reconsider all of the cases of the previous section, we see that by using the full Jacobian, we achieve the same virtual fixtures, albeit in a larger space of allowed motions. In particular, choosing $D = \langle J \rangle$ now prefers motions on a sphere about the observed point, together with rotation about and translation along, the line of sight. It is, however, important to note that the distance from the camera to the observed point no longer "drops out" of the system as in the case of pure translation. Therefore distance must be estimated, for example by using adaptive schemes as outlined in [26].

At this point, we redirect the reader to [3], where it is observed that regulating the motion of the camera through invariants defined on observed features or measures thereof creates a broad family of "virtual linkages" between the camera and the world. In effect, applying the constructions as laid out above to these control laws creates a corresponding family of virtual fixtures. Likewise, more recent "hybrid" approaches to control that seek to produce more reliable converge of visual servoing methods [23] and/or control other properties of the motion of features in the image

plane [4, 5] can be applied, to the extent that the properties outlined in Section 4.2.3 are satisfied.

4.3.3 More General Camera Configurations

Until now, we have only considered a single end-effector mounted camera. However, it is important to note that everything we have said above can be applied to the case of a fixed camera observing an independently moving manipulator, with suitable adjustment of the form of the image Jacobian. Furthermore, we note that for either configuration, observing both the end-effector and the external features defining the task creates an endpoint closed-loop control law which has well-known robustness against camera calibration error [8, 10, 13, 14]. Likewise, methods for estimating the image Jacobian on line [15] can, in principle, be applied practically without change.

As a final generalization, we could also add a second observing camera. It is well known [10, 14] that the relationship between control velocities and changes in observed image errors are expressed by "stacking" the individual image Jacobians for each camera, now expressed in a common coordinate system. Furthermore, the estimate of z (depth) in the Jacobian becomes trivial using triangulation from the two cameras. If we return to our list of examples, the following comments apply.

Pure translation. In the case of feature points and pure translation, the stacked Jacobian matrix spans the entire space of robot motions (except for points along the baseline of the two-camera systems), and therefore it is not possible to define interesting virtual fixtures other than point targeting.

If we consider the case of following a line, however, then when we stack the two Jacobians as before and apply the general vision-based virtual fixture rule, we arrive at a control law that effectively creates a prismatic joint that permits motion strictly along a line in space.

General motion. For general motion, we see that the "stack" of two Jacobians for feature-point servoing creates a spherical joint: the preferred DOF are motion on a sphere about the observed point while maintaining direction to the point (2 DOF) and rotation about that line of sight (1 DOF). This may, at first, seem counter-intuitive since the stacked Jacobian has 4 rows. However, due to the epipolar constraints of the camera, one of these constraints is redundant and thus the Jacobian spans only 3 DOF. If we add a second point, we further reduce the DOF by 2, with the remaining allowed motion being rotation about the line defined by the two observed points. A third observed point completely determines the pose of the observed (or observing) system, and so virtual fixturing once again reduces to motion to a target pose.

In the case of placing a point on a line, the image constraints now create a constraint on two positional DOF. It is, however, still possible to rotate in any direction (with the constraint that the rotation preserved distance to the observed point) and to move along the line. Placing a second point on the line reduces this to 2 DOF (rotation about the line and translation along it).

For a complete categorization of image-plane constructions for two-camera systems, we refer the reader to [7, 10].

4.4 Examples of Use

In our initial work, a preliminary version of the algorithms described above were implemented on the JHU SHR [19]. Here, we briefly describe the setup, the results, and its relationship to the more general framework given above. More details can be found in [1].

The robot was equipped with a vision sensor rigidly attached to the force-sensing handle on the end effector. We chose to execute two-dimensional tasks parallel to the image plane, which was in turn arranged to be parallel to two of the base stages of the robot. We performed experiments using a charge-coupled device (CCD) camera at the macro scale and a GRIN lens endoscope at the micro scale. The vision sensor always viewed the task plane, allowing reading of the motion references and real-time display of task execution (Figure 4.1). On-screen display of the task execution is useful for operators at the macro scale and essential at the micro scale, as it would be impossible to complete the task using the naked eye.

The path was furnished to both the system and the user by printing a sine curve (35 mm amplitude, 70 mm wavelength, and 0.54 mm width) on the task plane (in black on white paper). At micro scale, it was not possible to print a sufficiently smooth curve, so we instead embedded a wavy human hair (about 80 μm diameter) in glue on a yellow sheet of paper. In the macro case, the camera was positioned 200 mm from the paper, yielding a pixel footprint of 0.066 mm on the working surface. In the micro case, the endoscope was about 150 μ m above the working surface, yielding a pixel footprint of about 1 μ m (Figure 4.2).

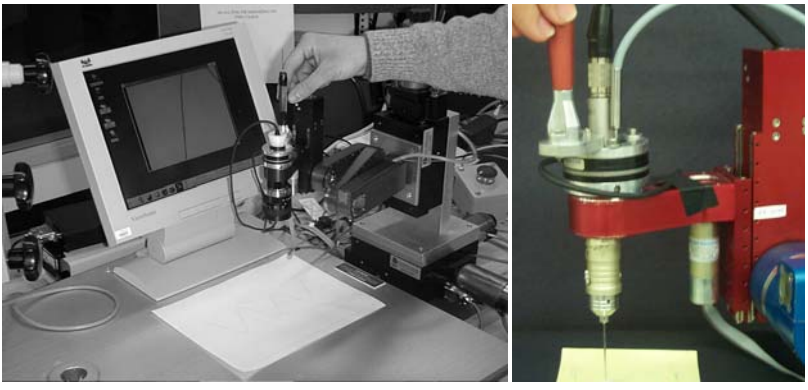
The center of the image was graphically marked, and users were instructed to perform path following tasks relative to this mark. The sensor on the handle was used to record user commands. The force sensor resolution is 12.5 mN and force values are expressed as multiples of this base unit.

Visual tracking (XVision system [11]) was used to measure (in real time) the local position and tangent direction, $\mathbf{d}$, to the path. Subpixel interpolation was used to increase the precision of these measurements. The vision and control subsystems executed on two different personal computers (PCs), and the data exchange was realized over a local network. The control system operated at 100 Hz, using the most recent available data from the vision system and handle force sensor.

In terms of our previous formulation, the marked image location at the image center means that $\mathbf{x} = 0$ Further, the workspace is a plane, $(\mathbf{x} \in \Re^2)$. The preferred direction is given by the tangent measurements from the tracking algorithm. Implicitly, the control law used to position the manipulator on the line is

$$\mathbf{u} = \mathbf{s} - \mathbf{x} = \mathbf{s}, \tag{4.22}$$

where $\mathbf{s} \in \Re^2$ is the current location of the visual tracker in the image. Further, $\mathbf{s}$ was constrained to lie along the line through the marked image location, normal



(a) (b)

Fig. 4.1 The experimental setup of the JHU SHR used to evaluate virtual fixtures to assist in planar path following and positioning tasks: (a) macro scale; and (b) micro scale

to **d**. Choosing **d** as the preferred direction of motion, we see the conditions of the general virtual fixturing rule are satisfied.

This class of virtual fixtures has been tested at both macro and micro scales. Results for a specific user and a wide class of compliances and situations can be found in [1, 2]. Tests for a larger class of users can be found in [24].

We have also developed and evaluated several concrete applications of the virtual fixture methods described in this chapter. In [22], we have applied this approach to virtual fixture guidance to the problem of fine lead soldering for small lot electronics manufacturing (Figure 4.3(a)). There, we show that novices improve at fine lead soldering using virtual fixturing methods. As noted earlier, retinal surgery has been the main focus of this work (Figure 4.3(b)). In [6], we developed a stereo vision-based guidance system for retinal surgery, and showed improved accuracy using virtual fixtures. Finally, we have also investigated modifications of the virtual fixture

Fig. 4.2 Endoscopic image of the 80 μ m-diameter human hair used as the path in micro scale experiments

(a) (b)

Fig. 4.3 Two applications of virtual fixture: (a) the InvenioSix high-precision robot in a configuration to support fine lead soldering; and (b) the latest generation SHR in a configuration designed for eye surgery

paradigm to guide dynamic motion on both admittance and impedance style robots [27], and to better accommodate dynamics induced by the human user [25].

4.5 Conclusion

In this chapter, we have outlined a broad theory of compliant virtual fixtures, and have applied that theory to the specific case of vision-guided assistance. Our earlier work suggests that such virtual fixtures can be a useful aid to dexterous manipulation.

In many ways, this chapter is intended to point toward interesting further directions to be explored. First and foremost, we have begun to develop a general means for translating control algorithms into virtual fixtures. However, the treatment to this point has not been sufficiently formal to determine when such a translation is possible. In particular, we have not described how to exhibit a set of preferred directions that are consistent with a control input. Further, we have not offered a formal definition of guidance with virtual fixtures that would permit a general theoretical statement of an equivalence between active control and passive virtual fixturing. These remain interesting open problems.

On the practical side all of our experiments with vision-guided virtual fixtures have been within a very specific setup. Numerous issues must be solved before a robust, general implementation of vision-guided virtual fixtures can be achieved. For example, in our previous work, gain shaping was essential to maintain stability. Similarly, there needs to be careful gain shaping to accommodate the differing scales of forces and torques. More importantly, the ergonomics of this wider class of guidance modes remains to be explored.

Finally, it is important to point out that most virtual fixtures apply in a very limited task context. Thus, it is important to consider how to combine guidance modes

in parallel (*e.g.* a force-based guidance mode along a needle axis combined with a vision-based virtual fixture to position the needle and a position-based alignment fixture), and to sequence them. Ideas along these lines can be found in [17].

Acknowledgements. We would like to acknowledge the contributions of Panadda Marayong and Allison Okamura to this work. This material is based upon work supported by the National Science Foundation under Grant Nos. IIS-0099770 and EEC-9731478. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[1] Bettini, A., Lang, S., Okamura, A., Hager, G.: Vision assisted control for manipulation using virtual fixtures. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1171–1176 (2001)

[2] Bettini, A., Lang, S., Okamura, A., Hager, G.: Vision assisted control for manipulation using virtual fixtures: Experiments at macro and micro scales. In: Proc. IEEE International Conference on Robot. Automat., pp. 3354–3361 (2002)

[3] Chaumette, F., Rives, P., Espiau, B.: Classification and realization of the different vision-based tasks. In: Hashimoto, K. (ed.) Visual Servoing, pp. 199–228. World Scientific, Singapore (1994)

[4] Corke, P., Hutchinson, S.: A new partitioned approach to image-based visual servo control. IEEE Trans. Robot. Autom. 17(4), 507–515 (2001)

[5] Cowan, N., Weingarten, J., Koditschek, D.: Vision-based control via navigation functions. IEEE Trans. Robot. Autom. (2002) (to appear)

[6] Dewan, M., Marayong, P., Okamura, A., Hager, G.D.: Vision-Based Assistance for Ophthalmic Micro-Surgery. In: Proceedings of Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), vol. 2, pp. 49–57 (2004)

[7] Dodds, Z.: Task specification languages for uncalibrated visual servoing. Ph.D. thesis, Yale University (2000)

[8] Dodds, Z., Hager, G., Morse, A., Hespanha, J.: Task specification and monitoring for uncalibrated hand/eye coordination. In: Proc. IEEE Int. Conference Rob. Automat., pp. 1607–1613 (1999)

[9] Forsyth, D., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall, Englewood Cliffs (2002)

[10] Hager, G.D.: A modular system for robust hand-eye coordination. IEEE Trans. Robot. Automat. 13(4), 582–595 (1997)

[11] Hager, G.D., Toyama, K.: The "XVision" system: A general purpose substrate for real-time vision applications. Comp. Vision, Image Understanding 69(1), 23–27 (1998)

[12] Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision. Addison Wesley, Reading (1993)

[13] Hespanha, J., Dodds, Z., Hager, G.D., Morse, A.S.: What tasks can be performed with an uncalibrated stereo vision system? International Journal of Computer Vision 35(1), 65–85 (1999)

[14] Hutchinson, S., Hager, G.D., Corke, P.: A tutorial introduction to visual servo control. IEEE Trans. Robot. Automat. 12(5), 651–670 (1996)

[15] Jägersand, M.: On-line estimation of visual-motor models for robot control and visual simulation. Ph.D. thesis, University of Rochester (1997)
[16] Kim, D., Rizzi, A., Hager, G.D., Koditschek, D.: A "robust" convergent visual servoing system. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. I, pp. 348–353. IEEE Computer Society Press, Los Alamitos (1995)
[17] Kragic, D., Marayong, P., Li, M., Okamura, A., Hager, G.: Human-Machine Collaborative Systems for Microsurgical Applications. The International Journal of Robotics Research 24(9), 731–741 (2005)
[18] Kumar, R., Goradia, T.M., Barnes, A., Jensen, P.: Performance of robotic augmentation in microsurgery-scale motions. In: Taylor, C., Colchester, A. (eds.) MICCAI 1999. LNCS, vol. 1679, pp. 1108–1115. Springer, Heidelberg (1999)
[19] Kumar, R., Hager, G., Jensen, P., Taylor, R.H.: An augmentation system for fine manipulation. In: Proc. Medical Image Computing and Computer Assisted Intervention, pp. 956–965. Springer, Heidelberg (2000)
[20] Kumar, R., Jensen, P., Taylor, R.H.: Experiments with a steady hand robot in constrained compliant motion and path following. In: IEEE ROMAN, pp. 92–97 (1999)
[21] Lai, F., Howe, R.: Evaluating control modes for constrained robotic surgery. In: IEEE International Conference on Robotics and Automation, pp. 603–609 (2000)
[22] Lin, H., Marayong, P., Mills, K., Karam, R., Kazanzides, P., Okamura, A., Hager, G.D.: Portability and Applicability of Virtual Fixtures Across Medical and Manufacturing Tasks. In: Proc. of International Conference on Robotics and Automation (ICRA), pp. 225–230 (2006)
[23] Malis, E., Chaumette, F., Boudet, S.: 2 1/2 D visual servoing. IEEE Trans. Robot. Autom. 15(2), 238–250 (1999)
[24] Marayong, P., Bettini, A., Okamura, A.: Effect of virtual fixture compliance on human-machine cooperative manipulation. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2001) (to appear)
[25] Marayong, P., Hager, G., Okamura, A.: Control methods for guidance virtual fixtures in compliant human-machine interfaces. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008, pp. 1166–1172 (2008)
[26] Papanikolopoulos, N.P., Khosla, P.K.: Adaptive Robot Visual Tracking: Theory and Experiments. IEEE Trans. on Automatic Control 38(3), 429–445 (1993)
[27] Pezzementi, Z., Okamura, A., Hager, G.D.: Dynamic Guidance with Pseudoadmittance Virtual Fixtures. In: International Conference on Robotics and Automation (ICRA), pp. 1761–1767 (2007), /CIRL/publications/pdf/PezzementiICRA07.pdf
[28] Rosenberg, L.: Virtual fixtures. Ph.D. thesis, Dept. of Mechanical Engineering, Stanford University (1994)
[29] Strang, G. (ed.): Linear Algebra and its Applications. Academic Press, New York (1980)
[30] Taylor, R., Jensen, P., Whitcomb, L., Barnes, A., Kumar, R., Stoianovici, D., Gupta, P., Wang, Z.X.: Steady-hand robotic system for microsurgical augmentation. The International Journal of Robotics Research 18(12), 1201–1210 (1999)
[31] Weiss, J.: Injection of tissue plasminogen activator into a branch retinal vein in eyes with central retinal vein occlusion. Opthalmology 108(12), 2249–2257 (2001)

Chapter 5
Luminance: A New Visual Feature for Visual Servoing

Christophe Collewet and Eric Marchand

Abstract. This chapter is dedicated to a new way to achieve robotic tasks by 2D visual servoing. Contrary to most of related works in this domain where geometric visual features are usually used, we directly here consider the luminance of all pixels in the image. We call this new visual servoing scheme *photometric visual servoing*. The main advantage of this new approach is that it greatly simplifies the image processing required to track geometric visual features all along the camera motion or to match the initial visual features with the desired ones. However, as it is required in classical visual servoing, the computation of the so-called *interaction matrix* is required. In our case, this matrix links the time variation of the luminance to the camera motions. We will see that this computation is based on a illumination model able to describe complex luminance changes. However, since most of the classical control laws fail when considering the luminance as a visual feature, we turn the visual servoing problem into an optimization one leading to a new control law. Experimental results on positioning tasks validate the feasibility of photometric visual servoing and show its robustness regarding to approximated depths, Lambertian and non Lambertian objects, low textured objects, partial occlusions and even, to some extent, to image content.

5.1 Introduction

Visual servoing is now a widely used technique in robot control [4]. More generally, it consists of using information provided by a vision sensor to control the state of a dynamic system. Robust extraction and real-time spatio-temporal tracking of visual cues is then usually one of the keys to the success of a visual servoing task. We will show here that this tracking process can be totally removed and that no

Christophe Collewet and Eric Marchand
INRIA Rennes - Bretagne Atlantique, IRISA, Lagadic team, France
e-mail: {christophe.collewet,eric.marchand}@inria.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 71–90.
springerlink.com

other information than the image intensity (that is the pure luminance signal) can be considered to control the robot motion.

Classically, to achieve a visual servoing task, a set of visual features has to be selected from the image in order to control the desired degrees of freedom (DOF). A control law has also to be designed so that these visual features $\mathbf{s}$ reach a desired value $\mathbf{s}^*$, leading to a correct realization of the task. The control principle is thus to regulate to zero the error vector $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$. To build this control law, the interaction matrix $\mathbf{L_s}$ is required. For eye-in-hand systems, this matrix links the time variation of $\mathbf{s}$ to the camera instantaneous velocity $\mathbf{v}$

$$\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v} \tag{5.1}$$

with $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v}$ is the linear camera velocity and $\boldsymbol{\omega}$ its angular velocity. Thereafter, if we consider the camera velocity as input of the robot controller, the following control law is designed to try to obtain an exponential decoupled decrease of the error $\mathbf{e}$

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_{\mathbf{s}}^{+}\mathbf{e} \tag{5.2}$$

where λ is a proportional gain that has to be tuned to minimize the time-to-convergence, and $\widehat{\mathbf{L}}_{\mathbf{s}}^{+}$ is the pseudo-inverse of a model or an approximation of $\mathbf{L_s}$ [4].

As it can be seen, visual servoing explicitly relies on the choice of the visual features $\mathbf{s}$ (and then on the related interaction matrix); that is the key point of this approach. However, with a vision sensor providing 2D measurements $\mathbf{x}(\mathbf{r}_k)$ (where $\mathbf{r}_k$ is the camera pose at time k), potential visual features $\mathbf{s}$ are numerous, since 2D data (coordinates of feature points in the image, contours, moments,...) as well as 3D data provided by a localization algorithm exploiting $\mathbf{x}(\mathbf{r}_k)$ can be considered. In all cases, if the choice of $\mathbf{s}$ is important, it is always designed from the visual measurements $\mathbf{x}(\mathbf{r}_k)$. However, a robust extraction, matching (between $\mathbf{x}(\mathbf{r}_0)$ and $\mathbf{x}^* = \mathbf{x}(\mathbf{r}^*)$ where $\mathbf{r}^*$ is the camera desired pose) and real-time spatio-temporal tracking (between $\mathbf{x}(\mathbf{r}_{k-1})$ and $\mathbf{x}(\mathbf{r}_k)$) have proved to be a complex task, as testified by the abundant literature on the subject (see [17] for a recent survey on this subject). This image processing is, to date, a necessary step and considered also as one of the bottlenecks of the expansion of visual servoing. That is why some works tend to alleviate this problem. A first idea is to select visual features as proposed in [11, 14] or as in [19] to only keep visual features that are tracked with a high confident level (see also [7] where a more general approach is proposed). However, the goal of such approaches is not to simplify the image processing step but to take into account that it can fail. A more interesting way to avoid any tracking process is to use non geometric visual features. In that case, parameters of a 2D motion model are used as in [21, 24, 23, 8]. Nevertheless, such approaches require an important and complex image processing step. Removing the entire matching process is only possible when using directly the luminance as we propose.

Indeed, to achieve this goal we use as visual features the simplest feature that can be considered: the image intensity itself. We therefore call this new approach *photometric visual servoing*. In that case, the visual feature vector $\mathbf{s}$ is nothing but the image while $\mathbf{s}^*$ is the desired image. The error $\mathbf{e}$ is then only the difference

between the current and desired images (that is $\mathbf{e} = \mathbf{I} - \mathbf{I}^*$ where $\mathbf{I}$ is a vector that contains image intensity of all pixels).

However, considering the whole image as a feature has previously been considered [18, 9]. As in our case, the methods presented in [9, 18] did not require a matching process. Nevertheless they differ from our approach in two important points. First, they do not use directly the image intensity since an eigenspace decomposition is performed to reduce the dimensionality of image data. The control is then performed in the eigenspace and not directly with the image intensity. Moreover, this way to proceed requires the off-line computation of this eigenspace and then, for each new frame, the projection of the image on this subspace. Second, the interaction matrix related to the eigenspace is not computed analytically but learned during an off-line step. This learning process has two drawbacks: it has to be done for each new object and requires the acquisition of many images of the scene at various camera positions. Considering an analytical interaction matrix avoids these issues.

An interesting approach, which also consider the pixels intensity, has been recently proposed in [15]. This approach is based on the use of kernel methods that lead to a high decoupled control law. However, only the translations and the rotation around the optical axis are considered whereas, in our work, the 6 DOF are controlled. Another approach that does not require tracking nor matching has been proposed in [1]. It models collectively feature points extracted from the image as a mixture of Gaussian and try to minimize the distance function between the Gaussian mixture at current and desired positions. Simulation results show that this approach is able to control the 3 DOF of robot (and the 6 DOF under some assumptions). However, note that an image processing step is still required to extract the current feature points. Our approach does not require this step. Finally, in [2], the authors present an homography-based approach to visual servoing. In this method the image intensity of a planar patch is first used to estimate the homography between current and desired image which is then used to build the control law. Despite the fact that, as in our case, image intensity is used as the basis of the approach, an important image processing step is necessary to estimate the homography. Furthermore, the visual features used in the control law rely on the homography matrix and not directly on the luminance.

In the remainder of this chapter we first compute the interaction matrix related to the luminance in Section 5.2. Then, we reformulate the visual servoing problem into an optimization problem in Section 5.3 and propose a new control law dedicated to the specific case of the luminance. Section 5.4 shows experimental results on various scenes for several positioning tasks.

5.2 Luminance as a Visual Feature

The visual features that we consider here are the luminance I of each point of the image, that is

$$\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) = (\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \cdots, \mathbf{I}_{N\bullet}) \tag{5.3}$$

where $\mathbf{I}_{k\bullet}$ is nothing but the k-th line of the image. $\mathbf{I}(\mathbf{r})$ is then a vector of size $N \times M$ where $N \times M$ is the size of the image. As mentioned in Section 5.1, an estimation of the interaction matrix is at the center of the development of any visual servoing scheme. In our case, we have to derive the interaction matrix related to the luminance of a pixel in the image, that is

$$\lim_{dt \to 0} \frac{I(\mathbf{x}, t+dt) - I(\mathbf{x}, t)}{dt} = \mathbf{L}_I(\mathbf{x})\mathbf{v} \tag{5.4}$$

$\mathbf{x} = (x, y)$ being the normalized coordinates of the projection $\mathbf{p}$ of a point physical $\mathbf{P}$ belonging to the scene.

Before computing the interaction matrix $\mathbf{L}_I(\mathbf{x})$ in the general case, lets first consider the simpler case where the temporal luminance constancy hypothesis is assumed, as it is done in most of computer vision applications. Let us also assume that $\mathbf{p}$ has a small displacement $\mathbf{dx}$ in the time interval dt

$$I(\mathbf{x} + \mathbf{dx}, t + dt) = I(\mathbf{x}, t). \tag{5.5}$$

If $\mathbf{dx}$ is small enough, a first order Taylor series expansion of (5.5) around $\mathbf{x}$ can be performed yielding the so-called *optical flow constraint equation* (OFCE) [13]

$$\nabla I^\top \dot{\mathbf{x}} + I_t = 0 \tag{5.6}$$

with ∇I the spatial gradient of $I(\mathbf{x}, t)$[1] and $I_t = \partial I(\mathbf{x}, t)/\partial t$. Moreover, considering the interaction matrix $\mathbf{L}_\mathbf{x}$ related to $\mathbf{x}$ (*i.e.* $\dot{\mathbf{x}} = \mathbf{L}_\mathbf{x}\mathbf{v}$)

$$\mathbf{L}_\mathbf{x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{5.7}$$

(5.6) gives

$$I_t = -\nabla I^\top \mathbf{L}_\mathbf{x}\mathbf{v}. \tag{5.8}$$

However, note that I_t is nothing but the left part of (5.4). Consequently, from (5.4) and (5.8), we obtain the interaction matrix $\mathbf{L}_I(\mathbf{x})$ related to I at pixel $\mathbf{x}$

$$\mathbf{L}_I(\mathbf{x}) = -\nabla I^\top \mathbf{L}_\mathbf{x}. \tag{5.9}$$

Of course, because of the hypothesis required to derive (5.5), (5.9) can only be valid for Lambertian scenes, that is for surfaces reflecting the light with the same intensity in each direction. Besides, (5.9) is also only valid for a motionless lighting source with respect to the scene.

Indeed, to derive the interaction matrix in the general case, we have to consider a more realistic reflection model than the Lambert's one. The Lambert's model can only explain the behavior of non homogeneous opaque dielectric material [22]. It only describes a diffuse reflection component and does not take into account the

[1] Let us point out that the computation of ∇I is the *only* image processing step necessary to implement our method.

viewing direction. We propose to use the well-known Phong model [20]. However, note that this model is not based on physical laws, but comes from the computer graphics community. Although empirical, it is widely used thanks to its simplicity, and because it is appropriate for various types of materials, whether they are rough or smooth. Note that other models could be considered such as the Blinn-Phong [3] as reported in [5].

According to the Phong model (see Figure 5.1), the intensity $I(\mathbf{x})$ at point $\mathbf{x}$ writes as follows

$$I(\mathbf{x}) = K_s \cos^k \alpha + K_d \cos\theta + K_a. \tag{5.10}$$

This relation is composed of a diffuse, a specular and an ambient component and assumes a point light source. The scalar K_s describes the specular component of the lighting; K_d describes the part of the diffuse term which depends on the *albedo* in $\mathbf{P}$; K_a is the intensity of ambient lighting in $\mathbf{P}$. Note that K_s, K_d and K_a depend on $\mathbf{P}$. θ is the angle between the normal to the surface $\mathbf{n}$ in $\mathbf{P}$ and the direction of the light source $\mathbf{L}$; α is the angle between $\mathbf{R}$ (which is $\mathbf{L}$ mirrored about $\mathbf{n}$) and the viewing direction $\mathbf{V}$. $\mathbf{R}$ can be seen as the direction due to a pure specular object, where k allows to model the width of the specular lobe around $\mathbf{R}$, this scalar varies as the inverse of the roughness of the material.

In the remainder of this chapter, the unit vectors $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ correspond to the axis of the camera frame (see Figure 5.1).

Considering that $\mathbf{R}, \mathbf{V}$ and $\mathbf{L}$ are normalized, we can rewrite (5.10) as

$$I(\mathbf{x}) = K_s {u_1}^k + K_d u_2 + K_a \tag{5.11}$$

where $u_1 = \mathbf{R}^\top \mathbf{V}$ and $u_2 = \mathbf{n}^\top \mathbf{L}$. Note that these vectors are easy to compute, since we have

$$\mathbf{V} = -\frac{\tilde{\mathbf{x}}}{\| \tilde{\mathbf{x}} \|} \tag{5.12}$$

$$\mathbf{R} = 2u_2\mathbf{n} - \mathbf{L} \tag{5.13}$$

with $\tilde{\mathbf{x}} = (x, y, 1)$. In the general case, we consider the following dependencies

$$\begin{cases} \mathbf{V} = \mathbf{V}\big(\mathbf{x}(t)\big) \\ \mathbf{n} = \mathbf{n}\big(\mathbf{x}(t), t\big) \\ \mathbf{L} = \mathbf{L}\big(\mathbf{x}(t), t\big) \\ \mathbf{R} = \mathbf{R}\big(\mathbf{x}(t), t\big). \end{cases} \tag{5.14}$$

From the definition of the interaction matrix given in (5.4), its computation requires to write the total derivative of (5.11)

$$\dot{I} = k K_s u_1^{k-1} \dot{u_1} + K_d \dot{u_2}. \tag{5.15}$$

However, it is also possible to compute $\dot{I}$ as

$$\dot{I} = \nabla I^\top \dot{\mathbf{x}} + I_t = \nabla I^\top \mathbf{L}_{\mathbf{x}} \mathbf{v} + I_t \tag{5.16}$$

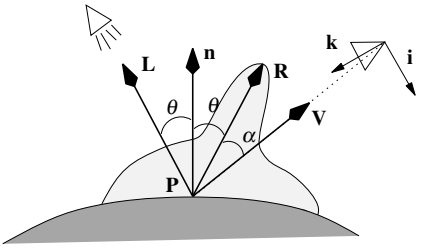


Fig. 5.1 The Phong illumination model [20]

where we have introduced the interaction matrix $\mathbf{L_x}$ associated to $\mathbf{x}$. Consequently, from (5.15) and (5.16), we obtain

$$\nabla I^\top \mathbf{L_x v} + I_t = kK_s u_1^{k-1} \dot{u_1} + K_d \dot{u_2} \tag{5.17}$$

that is a general formulation of the OFCE considering the Phong illumination model.

Thereafter, by explicitly computing the total time derivative of u_1 and u_2 and writing

$$\dot{u_1} = \mathbf{L}_1^\top \mathbf{v} \text{ and } \dot{u_2} = \mathbf{L}_2^\top \mathbf{v}, \tag{5.18}$$

we obtain the interaction matrix related to the intensity at pixel $\mathbf{x}$ in the general case

$$\mathbf{L}_I = -\nabla I^\top \mathbf{L_x} + kK_s {u_1}^{k-1} \mathbf{L}_1^\top + K_d \mathbf{L}_2^\top. \tag{5.19}$$

Note that we recover the interaction matrix $-\nabla I^\top \mathbf{L_x}$ associated to the intensity under temporal constancy (see (5.9)), *i.e.* in the Lambertian case ($K_s = 0$) and when $\dot{u_2} = 0$ (*i.e.* the lighting direction is motionless with respect to the point $\mathbf{P}$).

To compute the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ involved in (5.19) we have to explicitly express $\dot{u_1}$ and $\dot{u_2}$. However, to do that, we have to assume some hypothesis about how $\mathbf{n}$ and $\mathbf{L}$ move with respect to the observer. Various cases have been studied in [6]. Nevertheless, to make this chapter more readable, we report here only the case where the light source is mounted on the camera and only give the final equation. However, all the details can be found in [6].

In this case, considering a directional light source, we simply have $\mathbf{L} = -\mathbf{k}$. After tedious computations, it can be shown that

$$\mathbf{L}_2^\top = -\nabla n_z^\top \mathbf{L_x} + \mathbf{L}_4^\top \tag{5.20}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ and $\mathbf{L}_4^\top = \begin{bmatrix} 0 & 0 & 0 & n_y & -n_x & 0 \end{bmatrix}$.

$\mathbf{L}_1^\top$ is expressed as follows:

$$\mathbf{L}_1^\top = \left(\mathbf{V}^\top \mathbf{J^R} + \mathbf{R}^\top \mathbf{J^V}\right) \mathbf{L_x} + \mathbf{L}_3^\top \tag{5.21}$$

where $\mathbf{J^R}$ and $\mathbf{J^V}$ are respectively the Jacobian matrices related to $\mathbf{R}$ and $\mathbf{V}$ (see [6]) with respect to $\mathbf{x}$, while $\mathbf{L}_3^\top = \begin{pmatrix} 0 & 0 & 0 & \mathbf{L}_{3x} & \mathbf{L}_{3y} & \mathbf{L}_{3z} \end{pmatrix}$ with

$$\begin{cases} \mathbf{L}_{3x} = 2\big(\mathbf{n}^\top \mathbf{V}(\mathbf{n}\times\mathbf{k})^\top + \mathbf{k}^\top \mathbf{n}\big(\mathbf{n}\times\mathbf{V}\big)^\top\big)\mathbf{i} \\ \mathbf{L}_{3y} = 2\big(\mathbf{n}^\top \mathbf{V}(\mathbf{n}\times\mathbf{k})^\top + \mathbf{k}^\top \mathbf{n}\big(\mathbf{n}\times\mathbf{V}\big)^\top\big)\mathbf{j} \\ \mathbf{L}_{3z} = 2\mathbf{k}^\top \mathbf{n}\big(\mathbf{n}\times\mathbf{V}\big)^\top\mathbf{k}. \end{cases} \tag{5.22}$$

However, the interaction matrix is very often computed at the desired position [4]. Indeed, this way to proceed avoid to compute on-line 3D information like the depths for example. We also here consider this case. More precisely, we consider that, at the desired position the depth of all the points where the luminance is measured are equal to a constant value Z^*. That means that we consider that the object is planar and that the camera and the object planes are parallel at this position. This case is depicted on the Figure 5.2. Here, since we suppose that $\mathbf{J}^{\mathbf{n}} = \mathbf{0}$ and $\mathbf{n} = -\mathbf{k}$, it is straightforward to show that $\mathbf{L}_2^\top = \mathbf{0}$. Besides, since $\mathbf{n} = -\mathbf{k}$ and $\mathbf{L} = -\mathbf{k}$, we have $\mathbf{R} = -\mathbf{k}$. We also have $\mathbf{J}^{\mathbf{R}} = \mathbf{0}$. Consequently, from (5.21), $\mathbf{L}_1^\top$ becomes

$$\mathbf{L}_1^\top = -\mathbf{k}^\top \mathbf{J}^{\mathbf{V}} \mathbf{L}_{\mathbf{x}} + \mathbf{L}_3^\top \tag{5.23}$$

while $\mathbf{L}_3^\top$ writes $\begin{pmatrix} 0 & 0 & 0 & -2\mathbf{V}^\top\mathbf{j} & -2\mathbf{V}^\top\mathbf{i} & 0 \end{pmatrix}$. Finally, using explicitly $\mathbf{V}$, $\mathbf{J}^{\mathbf{V}}$ and $\mathbf{L}_{\mathbf{x}}$, we simply obtain

$$\mathbf{L}_1^\top = \frac{1}{\| \tilde{x} \|} \begin{bmatrix} \frac{x}{\bar{Z}} & \frac{y}{\bar{Z}} & -\frac{x^2+y^2}{\bar{Z}} & y & -x & 0 \end{bmatrix} \tag{5.24}$$

where $\bar{Z} = Z^* \| \tilde{\mathbf{x}} \|^2$.

As it can be seen, even if the computation of the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ to derive the interaction matrix is not straightforward, their final expression is very simple and easy to compute on-line.

5.3 Visual Servoing Control Law

The interaction matrix associated to the luminance being known, the control law can be derived. Usually it is based on a desired behavior for the error signal $\mathbf{e}$. More often, an exponential decoupled decrease of this signal is required, that is $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ where λ is a positive scalar. Therefore, expressing the temporal derivative of $\mathbf{e}$, we have

$$\dot{\mathbf{e}} = \mathbf{L}_{\mathbf{s}} \mathbf{v} = -\lambda \mathbf{e} \tag{5.25}$$

leading to the classical control law given in (5.2) when considering that only an approximation or an estimation of the interaction matrix is available.

However, we think that presenting the design of a control law from an optimization problem, as proposed in [16], can lead to more powerful control laws.

5.3.1 Visual Servoing as an Optimization Problem

In that case, the cost function that we have to minimize with respect to the camera current pose writes as follows

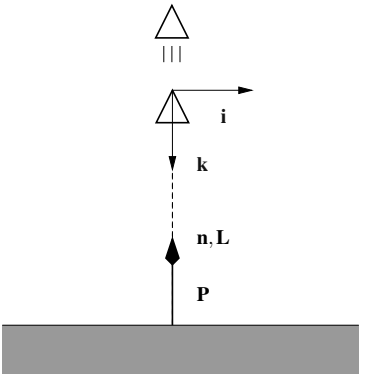


Fig. 5.2 Light source mounted on the camera for a planar object when the camera and the object planes are parallel

$$\mathscr{C}(\mathbf{r}) = \frac{1}{2} \parallel \mathbf{e} \parallel^2 \tag{5.26}$$

where $\mathbf{e} = \mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)$.

Nevertheless, regardless of the complexity of the shape of (5.26), since to evaluate (5.26) at a given pose a motion has to be performed, this problem becomes more complex than a classical optimization one if we want to ensure a suitable camera trajectory. Therefore, powerful approaches based on backstepping cannot be used. Indeed, in practice, only differential approaches can be employed to solve this particular optimization problem. In that case, a step of the minimization scheme can be written as follows

$$\mathbf{r}_{k+1} = \mathbf{r}_k \oplus t_k \mathbf{d}(\mathbf{r}_k) \tag{5.27}$$

where "$\oplus$" denotes the operator that combines two consecutive frame transformations; $\mathbf{r}_k$ is the current pose, t_k is a positive scalar (the descent step) and $\mathbf{d}(\mathbf{r}_k)$ a descent direction ensuring that (5.26) decreases if

$$\mathbf{d}(\mathbf{r}_k)^\top \nabla \mathscr{C}(\mathbf{r}_k) < 0. \tag{5.28}$$

Consequently, the following velocity control law can be easily derived considering that t_k is small enough

$$\mathbf{v} = \lambda_k \mathbf{d}(\mathbf{r}_k) \tag{5.29}$$

where λ_k is a scalar that depends on t_k and on the sampling rate. However, here again, since (5.26) cannot be simply evaluated or estimated, line search algorithms cannot be used and this value is often chosen as a constant one. In the remainder of this chapter we will omit the subscript k for the sake of clarity.

Several descent directions can be used, nevertheless they lead to the following generalized expression of (5.2) (see [6] for more details)

$$\mathbf{v} = -\lambda \widehat{\mathbf{N}_\mathbf{s}} \mathbf{e} \tag{5.30}$$

where:

- $\mathbf{N_s} = \mathbf{L_s}^\top$ for a steepest descent (gradient) method. For instance, this approach has been used in [12];
- $\mathbf{N_s} = \mathbf{L_s}^+$ for a Gauss–Newton (GN) method. It is the control law usually used. Note also that the case where $\mathbf{N_s} = \mathbf{L_{s^*}}^+$ is also very used in practice [10];
- $\mathbf{N_s} = \big(\mathbf{H} + \mu \operatorname{diag}(\mathbf{H})\big)^{-1}\mathbf{L_s}^\top$ for a Levenberg–Marquardt method. $\mathbf{H} = \mathbf{L_s}^\top\mathbf{L_s}$ is an approximation of the Hessian matrix of the cost function (see Section 5.3.2). The parameter μ makes possible to switch from a steepest descent like approach[2] to a GN one thanks to the observation of (5.26) during the minimization process; and
- $\mathbf{N_s} = (\mathbf{L_s} + \mathbf{L_{s^*}})^+$ for the efficient second order minimization (ESM) method proposed in [16]. Note that this method takes benefit of knowing the shape of the cost function near the global minimum (through $\mathbf{L_{s^*}}$); it is thus less sensitive to local minima than the above-mentioned methods. Its convergence domain is also larger.

In practice, since the convergence of the control law (5.30) highly depends on the cost function (5.26), we focus in the next section on its shape.

5.3.2 Shape of the Cost Function

In fact, we are interested in the shape of the cost function since we want to minimize it. Therefore, we are interested in studying the Hessian of (5.26). It is given by

$$\nabla^2 \mathscr{C}(\mathbf{r}) = \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right) + \sum_{i=1}^{i=\dim \mathbf{s}} \nabla^2 s_i \big(s_i(\mathbf{r}) - s_i(\mathbf{r}^*)\big). \tag{5.31}$$

However, this expression is far too complex to derive some useful results. Thus, we study it around the desired position $\mathbf{r}^*$, leading to

$$\nabla^2 \mathscr{C}(\mathbf{r}^*) = \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right). \tag{5.32}$$

Moreover, since we have $\dot{\mathbf{s}} = \dfrac{\partial \mathbf{s}}{\partial \mathbf{r}}\dot{r} = \mathbf{L_s}\mathbf{v}$, we are interested in practice in the following matrix

$$\mathbf{H}^* = \mathbf{L_{s^*}}^\top \mathbf{L_{s^*}}. \tag{5.33}$$

This matrix allows us to estimate the cost function around $\mathbf{r}^*$. Indeed, a first order Taylor series expansion of the visual features $\mathbf{s}(\mathbf{r})$ around $\mathbf{r}^*$ gives

$$\mathbf{s}(\mathbf{r}) = \mathbf{s}(\mathbf{r}^*) + \mathbf{L_{s^*}}\Delta\mathbf{r} \tag{5.34}$$

[2] More precisely, each component of the gradient is scaled according to the diagonal of the Hessian, which leads to larger displacements along the direction where the gradient is low.

where $\Delta\mathbf{r}$ denotes the relative pose between $\mathbf{r}$ and $\mathbf{r}^*$. Therefore, by plugging (5.34) into (5.26), we obtain an approximation of the cost function in a neighborhood of $\mathbf{r}^*$

$$\widehat{\mathscr{C}}(\mathbf{r}) = \frac{1}{2}\Delta\mathbf{r}^\top \mathbf{H}^* \Delta\mathbf{r}. \tag{5.35}$$

Of course, the graal would be that the eigenvalues of $\mathbf{H}^*$ are all equal since in that case the cost function would be an hypersphere. Indeed, only a global minimum would exist and a simple steepest descent method would ensure to reach this minimum. Unfortunately, when using the luminance as visual feature, the eigenvalues are very different[3]. On the other hand, the eigenvectors of $\mathbf{H}^*$ point out some directions where the cost function decreases slowly when its associated eigenvalue is low or decreases quickly when its associated eigenvalue is high. That means that the cost function (5.26) presents very narrow valleys. More precisely, an eigenvector associated to a small eigenvalue corresponds to a valley where the cost varies slowly. In contrast, the cost function varies strongly along an orthogonal direction. It can be shown that is in a direction near $\nabla\mathscr{C}(\mathbf{r})$ [6]. These preferential directions where the variation of the cost function is low are easy explained by the fact that it is very difficult to distinguish in an image an x axis translational motion (respectively y) from a y axis rotational motion (respectively x). The z axis being the camera optical axis.

5.3.3 Control Law

As shown in Section 5.3.1, several control laws can be used to minimize (5.26). We first used the classical control laws based on the GN approach and the ESM approach [16, 25]. Unfortunately, they may fail, either because they diverged or because they led to unsuitable 3D motion. It is well-known in optimization theory that minimizing a cost function that presents narrow valleys is a complex problem. Therefore, a new control law has to be derived.

We propose the following algorithm to reach its minimum. The camera is first moved to reach the valleys and next along the axes of the valleys towards the desired pose. It can be easily done by using a control law formally equal to the one used in the Levenberg–Marquardt approach (see Section 5.3.1). However, the way to tune the parameter μ is different. We denote this method in the remainder of the chapter as modified Levenberg–Marquardt (MLM). As stated in the Section 5.3.2, the first step can be easily done by using a gradient approach, that is by choosing a high value for μ (typically $\mu = 1$). Once the bottom of valleys has been reached (see [6] for more details), the parameters μ is forced to decrease to turn the behavior of the algorithm to a GN approach. The resulting control law is then given by

$$\mathbf{v} = -\lambda \left(\mathbf{H} + \mu \operatorname{diag}(\mathbf{H})\right)^{-1} \mathbf{L_I}^\top \mathbf{e} \tag{5.36}$$

where μ is not a constant value.

[3] Note that this phenomenon also holds for most of the geometrical visual features usually used in visual servoing since a term related to the depth always occurs in the translational part of the interaction matrix (see (5.7)).

5.4 Experimental Results

In all the experiments reported here, the camera is mounted on a 6 DOF gantry robot. Control law is computed on a Core[4] 2 Duo 3 GHz PC running Linux[5]. Images are acquired at 66 Hz using an IEEE 1394 camera with a resolution of 320 $\times$ 240. The size of the vector $\mathbf{s}$ is then 76800. Despite this size, the interaction matrix $\mathbf{L_I}$ involved in (5.36) can be computed at each iteration if needed.

5.4.1 Positioning Tasks under Temporal Luminance Constancy

We assume in this section that the temporal luminance constancy hypothesis is valid. To make this assumption as valid as possible, a diffuse lighting as been used so that $I(\mathbf{x})$ can be considered as constant with respect to to the viewing direction. Moreover, the lighting is also motionless with respect to the scene being observed. In this section, we will first compare the GN and MLM methods and then show that the photometric visual servoing is robust.

5.4.1.1 Comparison between the GN and the MLM Methods

The goal of the first experiment is to compare the control laws based on GN and MLM approaches when a planar object is considered (it is a photo). The initial error pose was $\Delta\mathbf{r}_{init} = (\ 5$ cm, -23 cm, 5 cm, -12.5 deg, -8.4 deg, -15.5 deg $)$. The desired pose was so that the object and charge-coupled device (CCD) planes are parallel at $Z = Z^* = 80$ cm. The interaction matrix has been computed at each iteration but assuming that all the depths are constant and equal to Z^*, which is of course a coarse approximation.

Figure 5.3(a) depicts the behavior of cost functions using the GN method or the MLM method while Figure 5.3(b) depicts the trajectories (expressed in the desired frame) when using either the GN or the MLM method. Figures 5.3(c–d) depict respectively the translation errors for the GN and MLM method while Figures 5.3(e–f) depict respectively the orientation errors for the GN and MLM method. The initial and final images are reported respectively on Figures 5.3(g–h). First, as it can be seen on Figure 5.3(a), both the control laws converge since the cost functions vanish. However, the time-to-convergence with the GN method is much higher than the one of the MLM method. The trajectory when using the GN method is also shaky compared to the one of the MLM method (Figure 5.3(b)). Compare also Figure 5.3(c) with Figure 5.3(d) and Figure 5.3(e) with Figure 5.3(f). The velocity of the camera when using the MLM method is smoother than when using the GN method (Figure 5.3(d) and Figure 5.3(c)). This experiment clearly shows that the MLM method outperforms the GN one. Note that in both cases the positioning errors is very low, for the MLM method we obtained $\Delta\mathbf{r} = (0.26$ mm, 0.30 mm, 0.03 mm, 0.02 deg,

[4] Core™ 2 Duo is a trademark of Intel Corporation in the U.S. and other countries. http://www.intel.com

[5] Linux® is a registered trademark of Linus Torvalds. http://als.linuxtoday.com

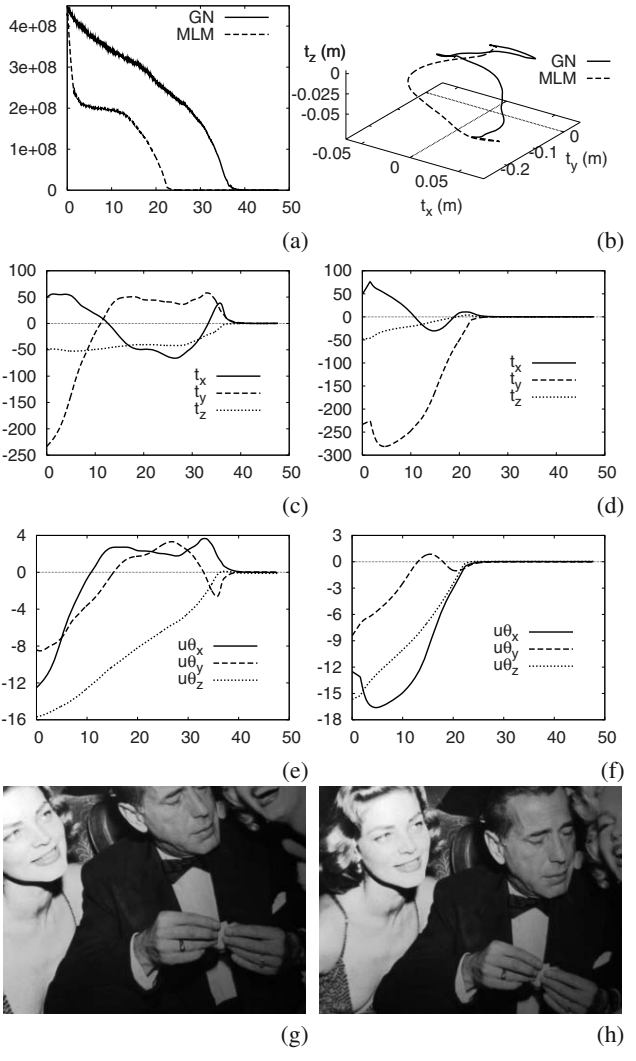


Fig. 5.3 First experiment, MLM versus GN method (x axis in seconds): (a) comparison of cost functions; (b) comparison of camera trajectories; (c) translation error for the GN method (in mm); (d) translation error for the MLM method (in mm); (e) rotation error for the GN method (in deg); (f) rotation error for the MLM method (in deg); (g) initial image; and (h) final image

-0.02 deg, 0.03 deg). It is very difficult to reach so low positioning errors when using geometric visual features as it is usually done. Indeed, these nice results are obtained because $\mathbf{I} - \mathbf{I}^*$ is very sensitive to the pose $\mathbf{r}$.

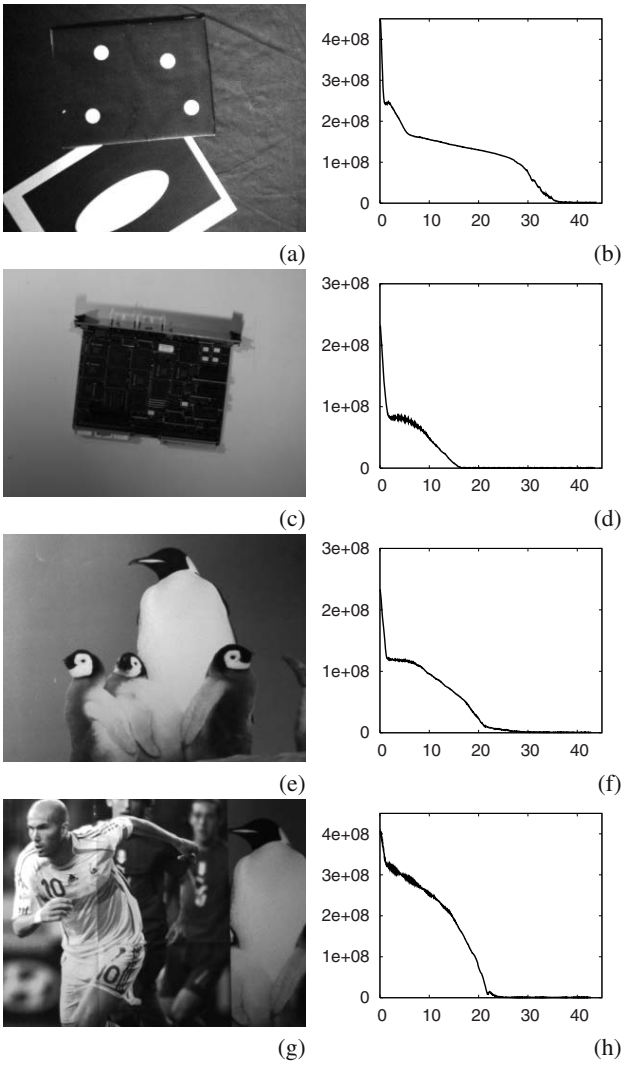


Fig. 5.4 Second experiment. Same positioning task with respect to various objects. Objects considered (left column) and cost functions (right column) (x axis in seconds)

5.4.1.2 Influence of the Image Content

The goal of the next experiment is to show that, even if the luminance is used as a visual feature, our approach does not depend too much on the texture of the scene being observed. Figure 5.4 depicts the behavior of our algorithm for the planar objects respectively shown on Figures 5.4(a,c,e,g) (the initial as well as the desired pose is unchanged). As it can be seen, the control law converges in each case, even in the case of a low textured scene (Figures 5.4(a–c)). Let us point out that

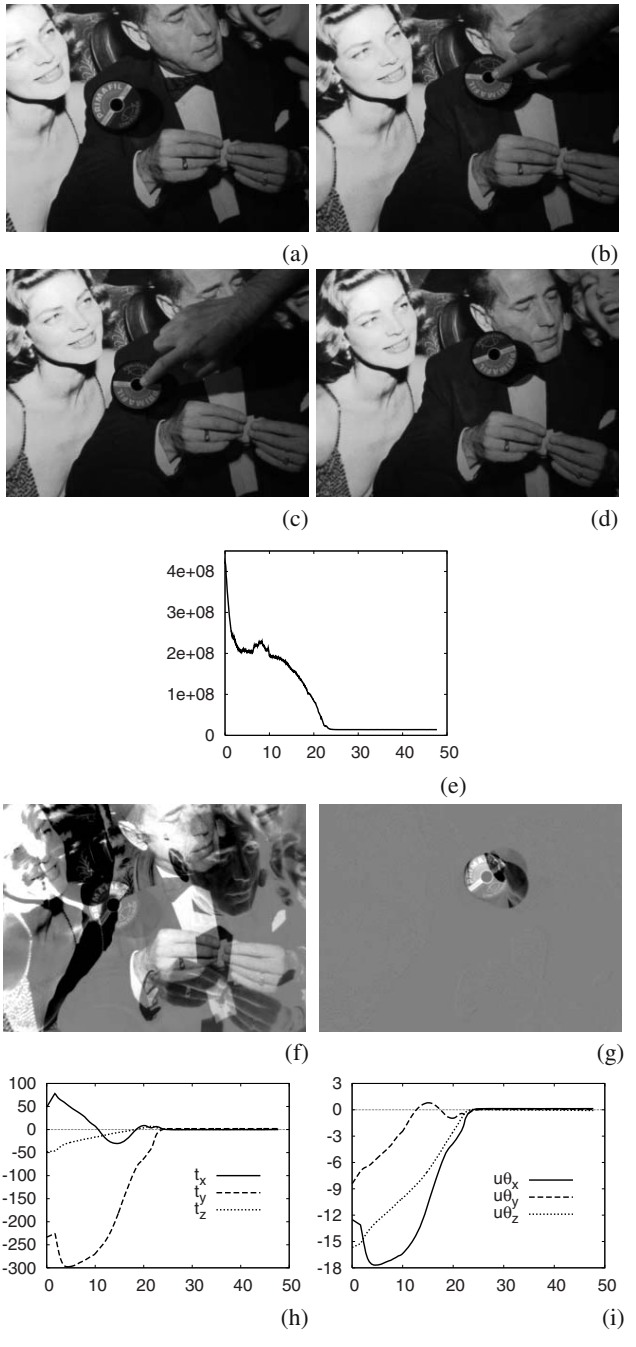


Fig. 5.5 Third experiment, occlusions (x axis in seconds): (a) initial image; (b) image at t $\approx$ 11 s; (c) image at t $\approx$ 13 s; (d) final image; (e) cost function; (f) $\mathbf{I} - \mathbf{I}^*$ at the initial position; (g) $\mathbf{I} - \mathbf{I}^*$ at the end of the motion; (h) translation errors (in mm); and (i) rotation errors (in deg)

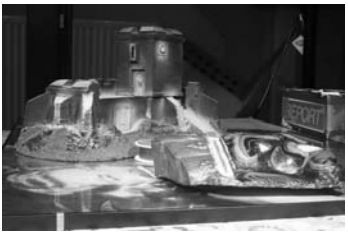


Fig. 5.6 The nonplanar scene

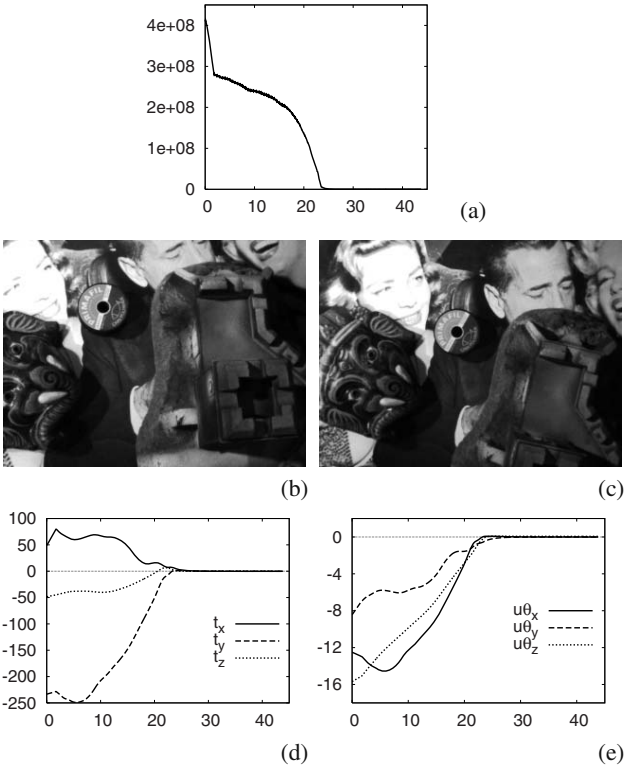


Fig. 5.7 Fourth experiment, robustness with respect to depths (x axis in seconds): (a) cost function; (b) initial image; (c) final image; (d) translation errors (in mm); and (e) rotation errors (in deg)

similar positioning errors than for the first experiment have been obtained. This result comes from the fact that the shape of the cost functions (5.26) does not depend too much on the image content (as long as the image does not contain periodic patterns or strong changes of the spatial gradient). It always presents narrow valleys that our control law can cope with.

Fig. 5.8 Camera and light-ring mounted on the robot end-effector

5.4.1.3 Behavior with respect to Partial Occlusions

The third experiment deals with partial occlusions. The desired object pose as well as the initial pose are still unchanged. After having moved the camera to its initial position, an object has been added to the scene, so that the initial image is now the one shown in Figure 5.5(a) and the desired image is still the one shown in Figure 5.3(h). Moreover, as seen in Figures 5.5(b–c), the object introduced in the scene is also moved by hand during the camera motion which highly increases the occluded surface. Despite that, the control law still converges. Of course, since the desired image is not the true one, the error cannot vanish at the end of the motion (see Figure 5.5(e)). Nevertheless, the positioning error is not affected by the occlusions (see Figures 5.5(h–i)) since the final positioning error is $\Delta\mathbf{r}$ = (-0.1 mm, 2 mm, 0.3 mm, 0.13 deg, 0.04 deg, 0.07 deg). It is very similar with the previous experiments. Compare also Figure 5.3(d) with Figure 5.5(h) and Figure 5.3(f) with Figure 5.5(i), the positioning error, and thus the camera trajectory, are really not affected by the occlusions. This very nice behavior is due to the high redundancy of the visual features we use.

5.4.1.4 Robustness to the Depths

The goal of the last experiment is to show the robustness of the control law with respect to the depths. For this purpose, a non planar scene has been used as shown on Figure 5.6. It shows that large errors in the depth are introduced (the height of the castle tower is around 30 cm). The initial and desired poses are unchanged. Figure 5.7 depicts this experiment. Here again, the control law still converges (despite the fact that the interaction matrix has been estimated at a constant depth Z^* = 80 cm) and the positioning error is still low since we have $\Delta\mathbf{r} = (0.2$ mm, -0.0 mm, 0.1 mm, -0.01 deg, 0.00 deg, 0.06 deg).

5.4.2 Positioning Tasks under Complex Illumination

In this section we consider the more complex case when the temporal luminance constancy is no more valid. Indeed, the scene is no more illuminated by a diffuse lighting since a directional light-ring is located around the camera lens

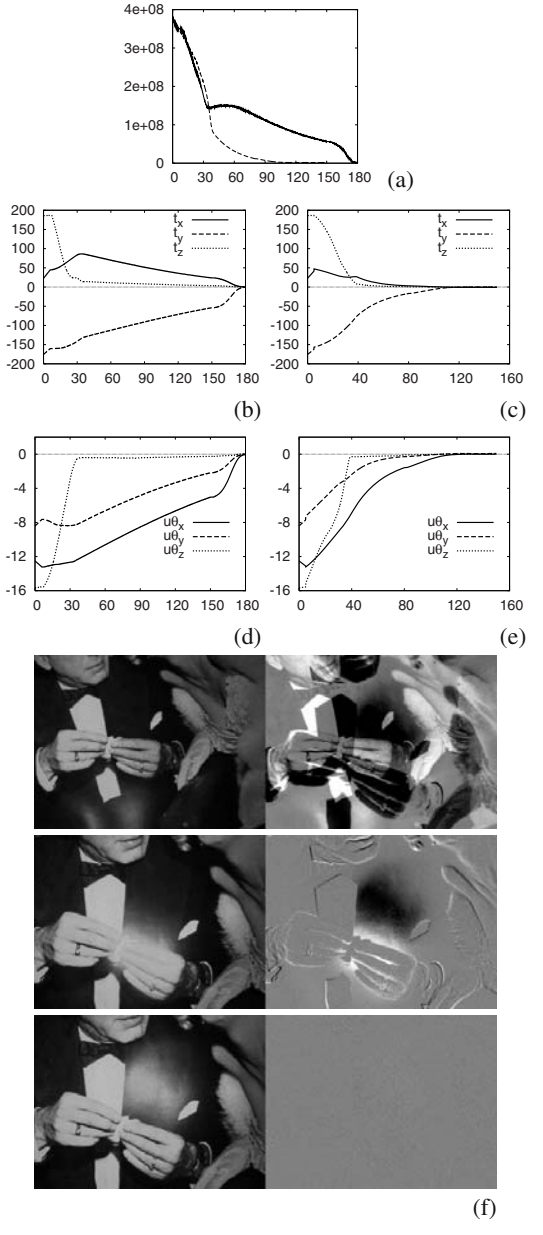


Fig. 5.9 Positioning task with the light source mounted on the camera: (a) cost function assuming a temporal luminance constancy model (solid line) and using an illumination model (dashed line); (b) translation error assuming a temporal luminance constancy model (in mm); (c) translation error using an illumination model (in mm); (d) rotation error assuming a temporal luminance constancy model (in deg); (e) rotation error using an illumination model (in deg); and (f) images acquired during the positioning task (left) and $\mathbf{I} - \mathbf{I}^*$ (right)

(see Figure 5.8). Therefore the light direction is aligned with the camera optical axis as described on Figure 5.2. This is the unique light in the scene. Note that, obviously, its direction is no more constant with respect to the scene as in Section 5.4.1. The initial positioning error and the desired pose are still unchanged (but with Z^*= 70 cm). The interaction matrix has been estimated at the desired position using (5.24) to compute $\mathbf{L}_1^\top$ while $\mathbf{L}_2^\top = 0$ (see the very end of Section 5.2). For all the experiments using the complete interaction matrix we used $k = 100$ and $K_s = 200$ (see (5.19)).

As it can be seen on Figure 5.9(f), the specularities are very important and consequently their motions in the image are important (for example the specularity can be seen at the bottom of the image in the first image whereas it has moved to the middle at the end of the positioning task). It also almost saturates the image meaning that few information are available around the specularity. The behavior of the control law is better when the complete illumination model is considered since the convergence is faster (see Figure 5.9(a)). It is also confirmed by observing the positioning errors (compare Figure 5.9(b) with Figure 5.9(c) and Figure 5.9(d) with Figure 5.9(e)).

Note that tracking tasks and other positioning tasks (when the lighting is not mounted on the camera) have been considered in [6]. These results show, here again, the benefit of using a complete illumination model instead of using the classical temporal luminance constancy.

5.5 Conclusion and Future Works

We have shown in this chapter the benefit of using the photometric visual servoing. This new visual servoing scheme avoids complex image processing, leaving only the image spatial gradient to compute. It also avoids a learning step required with previous approaches based on the use of the image intensity as visual features. This new visual servoing has also other important advantages. Concerning positioning tasks, the positioning errors are always very low. Moreover, this approach is not sensitive to partial occlusions and to coarse approximations of the depths required to compute the interaction matrix. Let us point out that the behavior of the robot is not disturbed by complex illumination changes since the interaction matrix has been derived from a suitable illumination model.

Future work will concern the case when the intensity of the lighting source may vary during the servoing.

Acknowledgements. The authors wish to thank François Chaumette and Seth Hutchinson for their constructive comments.

References

[1] Abdul Hafez, A., Achar, S., Jawahar, C.: Visual servoing based on gaussian mixture models. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2008, Pasadena, California, pp. 3225–3230 (2008)

[2] Benhimane, S., Malis, E.: Homography-based 2d visual tracking and servoing. Int. Journal of Robotics Research 26(7), 661–676 (2007)
[3] Blinn, J.: Models of light reflection for computer synthesized pictures. In: ACM Conf. on Computer graphics and interactive techniques, SIGGRAPH 1977, San Jose, California, pp. 192–198 (1977), http://doi.acm.org/10.1145/563858.563893
[4] Chaumette, F., Hutchinson, S.: Visual servoing and visual tracking. In: Siciliano, B., Khatib, O. (eds.) Handbook of Robotics, ch. 24, pp. 563–583. Springer, Heidelberg (2008)
[5] Collewet, C., Marchand, E.: Modeling complex luminance variations for target tracking. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR 2008, Anchorage, Alaska, pp. 1–7 (2008)
[6] Collewet, C., Marchand, E.: Photometric visual servoing. Tech. Rep. No. 6631, INRIA (2008)
[7] Comport, A., Marchand, E., Chaumette, F.: Statistically robust 2D visual servoing. IEEE Trans. on Robotics 22(2), 415–421 (2006), http://dx.doi.org/10.1109/TRO.2006.870666
[8] Crétual, A., Chaumette, F.: Visual servoing based on image motion. Int. Journal of Robotics Research 20(11), 857–877 (2001)
[9] Deguchi, K.: A direct interpretation of dynamic images with camera and object motions for vision guided robot control. Int. Journal of Computer Vision 37(1), 7–20 (2000)
[10] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Trans. on Robotics and Automation 8(3), 313–326 (1992)
[11] Feddema, J., Lee, C., Mitchell, O.: Automatic selection of image features for visual servoing of a robot manipulator. In: IEEE Int. Conf. on Robotics and Automation, ICRA 1989, Scottsdale, Arizona, vol. 2, pp. 832–837 (1989)
[12] Hashimoto, K., Kimura, H.: LQ optimal and non-linear approaches to visual servoing. In: Hashimoto, K. (ed.) Visual Servoing, vol. 7, pp. 165–198. World Scientific Series in Robotics and Automated Systems, Singapore (1993)
[13] Horn, B., Schunck, B.: Determining optical flow. Artificial Intelligence 17(1-3), 185–203 (1981)
[14] Janabi-Sharifi, F., Wilson, W.: Automatic selection of image features for visual servoing. IEEE Trans. on Robotics and Automation 13(6), 890–903 (1997)
[15] Kallem, V., Dewan, M., Swensen, J., Hager, G., Cowan, N.: Kernel-based visual servoing. In: IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS 2007, San Diego, USA (2007)
[16] Malis, E.: Improving vision-based control using efficient second-order minimization techniques. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2004, New Orleans, vol. 2, pp. 1843–1848 (2004)
[17] Marchand, E., Chaumette, F.: Feature tracking for visual servoing purposes. Robotics and Autonomous Systems 52(1), 53–70 (2005), http://dx.doi.org/10.1016/j.robot.2005.03.009; Special issue on "Advances in Robot Vision". In: Kragic, D., Christensen, H. (eds.)
[18] Nayar, S., Nene, S., Murase, H.: Subspace methods for robot vision. IEEE Trans. on Robotics 12(5), 750–758 (1996)
[19] Papanikolopoulos, N.: Selection of features and evaluation of visual measurements during robotic visual servoing tasks. Journal of Intelligent and Robotic Systems 13, 279–304 (1995)
[20] Phong, B.: Illumination for computer generated pictures. Communication of the ACM 18(6), 311–317 (1975)

[21] Questa, P., Grossmann, E., Sandini, G.: Camera self orientation and docking maneuver using normal flow. In: SPIE AeroSense 1995, Orlando, Florida, USA, vol. 2488, pp. 274–283 (1995)
[22] Reichmann, J.: Determination of absorption and scattering coefficients for non homogeneous media. Applied Optics 12, 1811–1815 (1973)
[23] Santos-Victor, J., Sandini, G.: Visual behaviors for docking. Computer Vision and Image Understanding 67(3), 223–238 (1997)
[24] Sundareswaran, V., Bouthemy, P., Chaumette, F.: Exploiting image motion for active vision in a visual servoing framework. Int. Journal of Robotics Research 15(6), 629–645 (1996)
[25] Tahri, O., Mezouar, Y.: On the efficient second order minimization and image-based visual servoing. In: IEEE Int. Conf. on Robotics and Automation, Pasadena, California, pp. 3213–3218 (2008)

Chapter 6
Visual Servoing for Beating Heart Surgery

Wael Bachta, Pierre Renaud, Ezio Malis, Koichi Hashimoto, and Jacques Gangloff

Abstract. Off-pump coronary artery bypass grafting (CABG) is still a technically challenging procedure. The existing mechanical stabilizers used for local suppression of the heart excursion have demonstrated significant residual motion, which could lead to a lack of accuracy in the surgical task, particularly in a minimally invasive surgery (MIS) context. Robots can help the surgeon by actively compensating for the heart motion using visual servoing. Various sensors like endoscopic camera, ultrasound imaging or even magnetic resonance imaging (MRI) can be used to provide the feedback of the visual loop. Advanced control approaches like predictive, repetitive or robust control can enhance the compensation accuracy. They rely on a model that uses physiological inputs to predict the motion of the myocardium in real-time.

6.1 Introduction

For many patients, CABG is still the only solution for myocardium revascularization. CABG is one of the most common surgical intervention in Europe and especially in North America. According to the organization for economic cooperation and development (OECD) 2008 health database [33], 12 European countries reported in 2006 an average of approximately 50 CABG procedures per 100,000 in-patient population. In 2004, the United States reported 145 procedures per 100,000 in-patient population. So, improving the quality of this intervention by

Wael Bachta, Pierre Renaud, and Jacques Gangloff
LSIIT, University of Strasbourg, Bd Sebastien Brant, 67412 Illkirch, France
e-mail: {bachta,pierre.renaud}@lsiit.u-strasbg.fr,
jacques.gangloff@lsiit.u-strasbg.fr

Ezio Malis
INRIA, 2004 Route des Lucioles, 06902 Sophia Antipolis, France
e-mail: ezio.malis@sophia.inria.fr

Koichi Hashimoto
Tohoku University, Aoba-ku Aramaki Aza Aoba 6-6-01, Sendai, Japan
e-mail: koichi@ic.is.tohoku.ac.jp

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 91–114.
springerlink.com

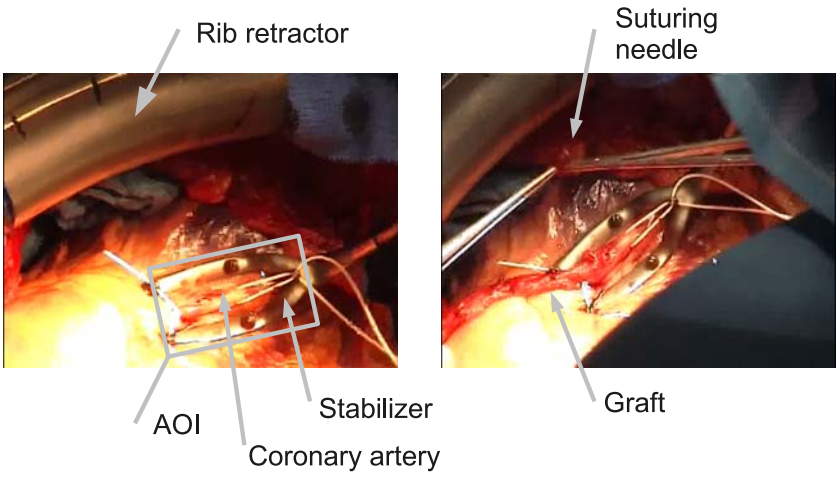


Fig. 6.1 Conventional off-pump CABG procedure

reducing complications, patient pain, hospital stay and recovery time would yield a major societal impact.

In a conventional CABG, access to the heart is obtained by a large cut in the chest along the sternum (sternotomy). The two parts of the rib cage are maintained thanks to a retractor (see Figure 6.1). When operating on a beating heart (off-pump), a mechanical stabilizer helps reducing the motion of the myocardium locally. The graft, previously harvested, is used to shunt the stenosed (abnormally narrowed) coronary artery. Connecting one side of the graft to the coronary artery requires many sutures: this connection is called the anastomosis.

One of the main way of improvement for CABG is minimally invasiveness. Using a minimally invasive access to the heart through tiny holes in the chest prevents from the complications due to sternotomy, reduces patient pain, hospital stay and recovery time. CABG performed this way is usually called totally endoscopic coronary artery bypass (TECAB). The other important improvement is beating heart surgery. Letting the heart beat during CABG or TECAB prevents from some serious complications, mainly neurological ones [8, 43]. This is now routinely performed in open-surgery but is still challenging in a minimally invasive context. The robot is a promising tool to help this kind of surgery. Thanks to telemanipulated endoscopic tools with internal mobilities and stereo visual feedback, the surgeon can achieve the grafting even on a moving organ. However, to reduce the motion of the area of interest (AOI) (approximately 1 cm^2 around the grafting area), surgeons have to use a heart stabilizer. It constrains mechanically and passively the relative motion between the AOI and the operation table. Despite this stabilization, the residual motion of the AOI is still significant [27] and surgeons cannot use the same techniques for TECAB than for open CABG, especially when suturing [18]. These workarounds to cope with residual motion have drawbacks and limitations [21].

So there is a real need to reduce the residual motion of the AOI. This can be obtained by canceling the relative motion between the AOI and the operation table or by canceling the relative motion between the instrument and the heart. Both approaches have been investigated in the literature. They will be detailed in Section 6.2. Most of these techniques involving a robotic device use visual servoing. Indeed, in a minimally invasive context, the endoscopic video of the AOI is always available. So it is relevant to use it as a feedback in a visual loop to regulate the relative motion AOI-table or AOI-instrument towards zero.

in vivo images are known to be very difficult to process: specularities due to the bright endoscopic light, large motions due to the high magnification factor and the lack of structure make the feature extraction process especially difficult. Robustness of the visual feedback is critical in order to meet the high safety standards for medical devices. Redundancy of sensors can improve the robustness: fusing the visual information with other signals can help maintaining a continuity of the feedback during occlusions for example.

The analysis of the heart motion reveals that its motion has some characteristic features [41]. This information can be used to make a predictive model of the AOI future displacements. A good motion prediction can drastically improve the precision of the visual servo loop thanks to anticipated control signals [19]. A description of the most recent prediction algorithms for local myocardium motion is given in Section 6.3.

The surface of the heart is deformable with few landmarks: only the coronary arteries network is visible. The vision algorithm should cope with this relatively poor visual information. Tracking algorithm based on novel pattern matching techniques have demonstrated their efficiency on live *in vivo* video of the myocardium. This point will be discussed in details in Section 6.4.

Robust vision and efficient motion prediction are not the only keys to achieve efficient visual servoing on the beating heart. The third key is dynamic control. Indeed, heart motion dynamics are fast [13]. Low frequency visual servoing is unable to cope with the sharp accelerations observed on the surface of the myocardium [13]. Furthermore, medical robots are lightweight and thus prone to flexibilities. Kinematic visual servoing [17] neglects these effects and so cannot achieve the optimal bandwidth required by the task. In Section 6.5 the synthesis of advanced control laws for efficient dynamic control is thus presented. Dynamic modeling of the visual loop is used in order to obtain the best performance. *in vivo* experimental results for stabilization and tracking tasks are given.

6.2 Motion Compensation Approaches

In the context of TECAB, the surgeon's hand should simultaneously reach high dexterity, to perform the anastomotic suture, and high dynamics to perform surgery on an organ that exhibits large displacements with large accelerations. The characteristics of the heart motion have been studied in details in the robotics community. On pigs, which heart has strong physiological similarities with the human

heart [12], the left anterior descending artery (LAD) may have an excursion of 12.5 mm [27], with accelerations up to 6 m/s^2 [13]. A spectral analysis of the motion shows that harmonics above 1 Hz (the approximate bandwidth limit of the human hand) are still significant. So, the surgeon cannot expect operating on the beating heart without a kind of motion compensation, even with a teleoperated robotic system like the *da Vinci*®[1].

Up to now, the only solution is to use stabilizers, to attenuate the heart motion. They indeed strongly limit the heart excursion around the anastomosis site, but the residual motion due to their flexibilities still makes the suturing task a delicate gesture, that only well-trained and experimented surgeons can perform. In the medical field, the lack of performances of mechanical stabilizers has been pointed out [10], notably in the case of totally endoscopic surgery [28]. Actually, in this latter case, a stabilizer presents a geometry that yields intrinsically significant residual motion: the long beam that constitutes the body of the device is deflected by the forces exerted by the heart.

Several approaches have therefore been proposed to overcome the lack of performances of mechanical stabilizers. They all consider the use of robotic systems to actively cancel the physiological motion, from the surgeon point of view. To do so, authors have considered either to synchronize the tool with the heart, virtually or physically, or to actively suppress the heart motion with respect to the operating table. These approaches are now presented and discussed.

6.2.1 Heart-tool Synchronization

Synchronization between the heart and the surgical tool is the most developed approach in the literature. In [42, 37], surgery is performed manually and the motion compensation is achieved through a moving platform that supports the surgeon hands. Thanks to a vision system mounted on the platform, the heart motion is measured and used for control. In the same time, a stabilized visual feedback of the tool and heart surface is provided to the surgeon. With such an approach, the surgeon tremor cannot be filtered. Moreover, such a collaborative approach does not seem to be compatible with a totally endoscopic procedure in the case of anastomosis. In totally endoscopic mitral valve surgery, a quite similar approach has been proposed recently [26]. However, the surgical task is limited to a tool translation along its axis that can be performed by the surgeon from the outside of the patient.

For TECAB procedures, synchronization should be considered in a teleoperation framework. Nakamura [31] introduced this concept, with a slave system positioned above the anastomosis site that tracks a point on the myocardium in order to virtually synchronize the tool and the heart thanks to high-speed visual servoing (955 Hz). Ginhoux *et al.* [19] extended this approach by using a predictive model of the heart motion. They demonstrated *in vivo* that visual servoing with predictive control can improve the tracking accuracy.

[1] da Vinci® Surgical System is a registered trademark of Intuitive Surgical, Inc. http://www.intuitivesurgical.com

With this approach, the slave robot is replacing the surgeon's hand. As a consequence, the dynamic properties of the robot must be compatible with the heart surface motion. Furthermore, the device must have enough intra-cavity mobilities to let the surgeon perform comfortably a suturing gesture. No commercial system, and even no research prototype have currently reach these severe requirements. From a mechanical design point of view, this is still a technical challenge.

Moreover, this approach may be questionable in terms of safety. Indeed, the robot kinetic energy constitutes a potential danger in case of control error or failure. Several safety principles are now widely accepted in the field of medical robot design [16]: minimizing the robot workspace, speed and acceleration is indeed a basic requirement. Safety has to be taken into account at the lowest design level, *i.e.* for the hardware design. One may consider that these principles are not respected in this case.

While remaining in the framework of heart/tool synchronization, an alternate approach could be the one adopted in the Heartlander project [35]. The robot is a miniature mobile robot directly positioned on the heart surface. It walks on the myocardium thanks to legs fitted with suction cups. The motion compensation is thus intrinsic to the system. However, the device has been designed initially to perform needle insertion on the epicardial surface. In the context of TECAB, design of mechanical structures at that scale with the force and kinematic requirements is an open problem.

6.2.2 Heart Immobilization

In this second approach, the surgical tool is not supposed to follow the heart surface. The idea is to remain in a teleoperation scheme, with two kinds of robotized laparoscopic tools. The first robotic system provides the needed intra-cavity mobilities, with a large workspace, and is teleoperated by the surgeon from a master console. The second robotic system stabilizes locally the heart surface: it is an active stabilizer that is designed to suppress the residual motion observed with conventional mechanical stabilizers.

With such a decomposition, we can limit the bandwidth of the slave system and in the same time minimize the workspace of the cardiac stabilizer, which means a quasi-suppression of its potential harmful effects. The project Cardiolock [3] aims at developing an active stabilizer compatible with MIS. Initially developed to compensate for residual displacements in a single direction, the Cardiolock device is now designed to perform a complete stabilization [4]. The required high accuracy and high dynamics have lead to the use of compliant architectures, with piezoelectric actuation. Backlash and friction are thus eliminated. High speed vision provides an exteroceptive feedback that allows the measurement of the stabilizer residual displacements.

In the following, the main properties of the device are introduced with the first Cardiolock prototype. This device globally consists of two parts. A first active part (on the left of Figure 6.2) is composed of a 1 degree of freedom (DOF) closed-loop

Fig. 6.2 CAD view of the first prototype of Cardiolock

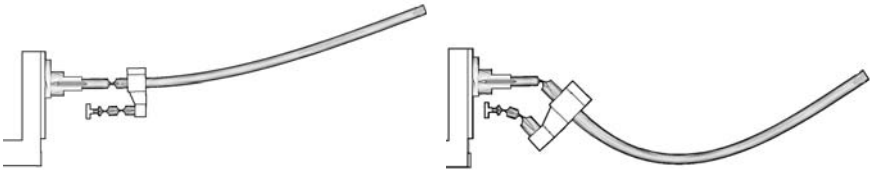


Fig. 6.3 Active compensation principle

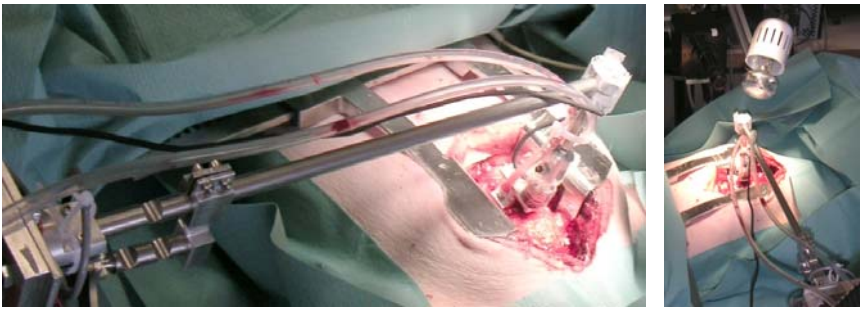


Fig. 6.4 Cardiolock *em vivo* experiments on pig

mechanism, which remains outside the patient body. The other part is a stainless steel beam which dimensions are compatible with MIS and that can be simply locked on the first part.

The asepsy of the device can be easily ensured: the external subsystem can be wrapped in a sterile bag and the other part can be sterilized using an autoclave.

In Figure 6.3 the principle of the device is introduced with a decomposition of the compensation task for illustration purpose. The left part shows the magnified deflection of the stabilizer due to the cardiac force; the stabilizer tip displacement is measured with vision. The right part shows the cancelation of the tip displacement by the modification of the closed-loop mechanism geometry.

One may notice that the approach developed in the Cardiolock project is quite similar to the one adopted in the MiCRoN project [11] for retinal surgery. This device is designed to compensate for the surgeon tremor. Only small displacements are achievable, with high dynamics. If we consider the classification introduced in [14],

these devices can be seen as belonging both to the category of surgical mechatronic tools. In that sense, the Cardiolock project is a novel and promising approach for beating heart surgery.

Experimental *in vivo* results (Figure 6.4) have shown the efficiency of the approach. Furthermore, the analysis of the contact forces between the stabilizer and the heart surface shows that the use of an active device does not modify significantly the pressure applied on the heart. Taking also into account the safety of the approach, we think that the medical acceptance of such a device should therefore be comparable to the one of currently used passive stabilizers.

6.3 Heart Motion Prediction

The cardiac motion is composed of two quasi-periodic components [41]. The slow component is due to the ventilation and the faster one corresponds to the heartbeats. The sharp transients of the heart motion correspond to the high frequency harmonics of the heartbeat. The heartbeat component has been proved to be dependent on the respiratory cycle [13]. As robotized heart compensation can take advantage of the knowledge of future cardiac motion, a lot of work has been devoted to heart motion prediction. A full review of the available prediction algorithms is given in [2]. In this section we only outline the two methods that explicitly take into account the coupling between the heartbeat and respiratory components.

6.3.1 Linear Parameter Varying Method

The first step of this prediction algorithm [13] consists in separating the heartbeat and respiratory components of the heart motion. This is achieved thanks to a gating technique. In the electrocardiogram (ECG) signal, the QRS complexes, a combination of three electric waves, denote the beginning of the cardiac cycle. Sampling the heart motion by the QRS complexes delayed by half a cardiac cycle allows to observe only the respiratory component at different moments of the respiratory cycle, since the heart is almost at rest during the second half of the heartbeat cycle. In order to be able to reconstruct the respiratory motion at each sample k, these samples are interpolated using a smoothing cubic spline function $\mathscr{F}$. Therefore, the respiratory motion can be written as

$$\hat{\mathscr{M}}_r(k) = \mathscr{F}(k - k_{resp}) \tag{6.1}$$

where k_{resp} is the sample number representing the beginning of the current respiratory cycle. Then, this motion is subtracted from the whole motion to obtain the heartbeat component:

$$\hat{\mathscr{M}}_c(k) = \mathscr{M}(k) - \hat{\mathscr{M}}_r(k). \tag{6.2}$$

The heartbeat component is modeled by a finite impulse response (FIR) to the QRS complex considered as an impulse. The heartbeat motion at sample k can therefore be written as a convolution product:

$$\hat{\mathscr{M}}_c(k) = l(k) * \mathrm{QRS}(k) \tag{6.3}$$

where $l(i)\,, i = 1 \ldots T_c$ are the coefficients of the FIR filter and T_c is the heartbeat period expressed in number of samples. In order to express the dependence of the heartbeat component on the respiratory cycle, the coefficients of the filter representing $\mathscr{M}_c$ are considered varying, and linearly dependent on the lung volume $\mathscr{V}(k)$:

$$l(k) = l_0(k) + l_1(k)\mathscr{V}(k). \tag{6.4}$$

The filter coefficients l_0 and l_1 are identified online using a recursive least-squares (RLS) algorithm. The future heart motion is predicted using this model and the periodicity of the lung volume ensured by the artificial ventilation. The expected motion at sample $k+n$ can be written as

$$\begin{aligned}\hat{\mathscr{M}}(k+n) &= \hat{\mathscr{M}}_r(k+n) + \hat{\mathscr{M}}_c(k+n) \\ &= \mathscr{F}(k-k_{resp}+n) + l_0(k-k_{QRS}+n) + l_1(k-k_{QRS}+n)\mathscr{V}(k-T_r+n)\end{aligned} \tag{6.5}$$

where k_{QRS} is the sample number representing the last QRS occurrence.

6.3.2 Amplitude Modulation Method

In [13], the authors modeled the coupling between the two motion components in the temporal domain. In this section we present an alternate method consisting in taking into account this coupling in a simpler way. Our algorithm can be seen as a generalization of the Fourier linear combiner (FLC) framework [41]. The proposed method is mainly based on experimental observations. Indeed, if we zoom in the spectral analysis of the heart motion we obtain plots comparable to Figure 6.5. The peaks around the heartbeat harmonics are similar to those that can be obtained by plotting the frequency content of an amplitude modulation of two periodic signals of frequencies f_c (the frequency of the carrier) and f_r (the frequency of the modulating signal).

Hence, we propose to write the heartbeat component of the heart motion as the result of an amplitude modulation. Two experimental observations should however be underlined. First, the number of harmonics in the modulating signal is lower than the number of harmonics in the respiratory component. These harmonics also exhibit different amplitudes. Second, only the first low frequency heartbeat harmonics are modulated. Therefore, we propose to write the heart motion as

$$\mathscr{M}(k) = \mathscr{M}_r(k) + \underbrace{\mathscr{C}_c(k) + \mathscr{C}_{c1}(k)\mathscr{C}_r(k)}_{\mathscr{M}_c(k)}. \tag{6.6}$$

$\mathscr{M}_r$ is the respiratory component of the heart motion. $\mathscr{C}_c$ contains all the significant heartbeat harmonics whereas $\mathscr{C}_{c1}$ contains only the first low frequency harmonics (a truncated part of $\mathscr{C}_c$). $\mathscr{C}_r$ is the modulating respiratory component. It has the same

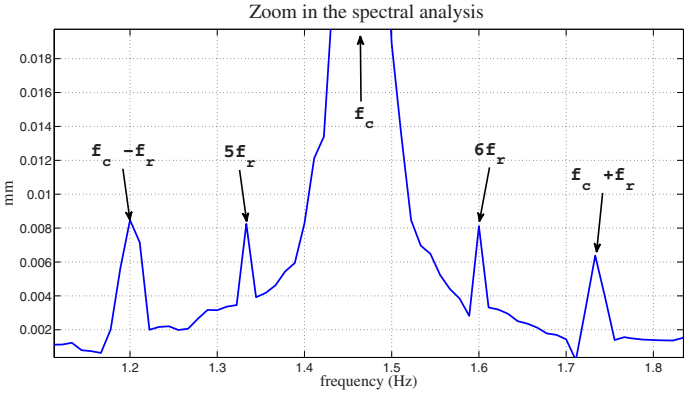


Fig. 6.5 A zoom in a spectral analysis of the heart motion

fundamental frequency than $\mathscr{M}_r$ but is different in amplitude and has a more limited bandwidth. Note that $\mathscr{C}_c(k)$ ensures the presence of the carrier in the modulated signal to fit the spectrum in Figure 6.5 (peak at f_c Hz).

The equation above can be written after a linearization as

$$\mathscr{M}(k) = \Phi^T(k)\,\mathbf{W}(\mathrm{k}), \tag{6.7}$$

where $\Phi^T(k)$ is the parameters vector that can be identified and updated online and $\mathbf{W}(\mathrm{k})$ is a regressor vector depending only on the current cardiac and respiratory states. As the future cardiac and respiratory states can be known in advance, future motion at sample $k+n$ can be simply computed as

$$\mathscr{M}(k+n) = \Phi^T(k+n)\,\mathbf{W}(\mathrm{k}). \tag{6.8}$$

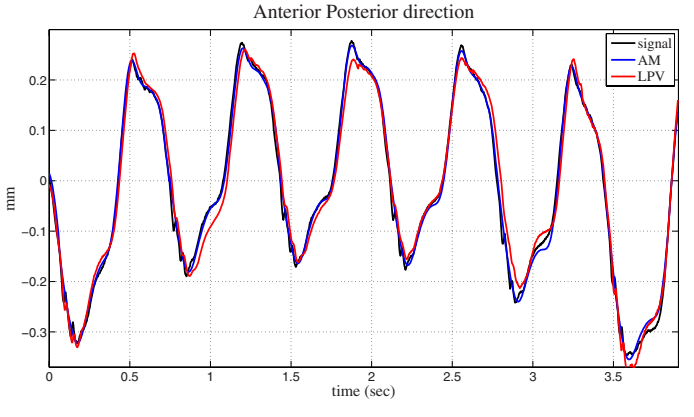


Fig. 6.6 One heartbeat period ahead prediction

Figure 6.6 shows one heartbeat ahead prediction results for the linear parameter varying (LPV) and amplitude modulation (AM) methods. The data used to make the assessment correspond to a pig heart motion in the anterior posterior direction, acquired during *in vivo* experiments. During these experiments the pig heart was restrained using a passive cardiac stabilizer. The prediction errors computed during 5 respiratory cycles are 25 μm and 13 μm respectively for the LPV and the AM methods. These results are satisfactory as they are below the surgical accuracy evaluated to 100 μm [3]. We should however keep in mind that the prediction algorithm is only one of the keys necessary to build an efficient predictive control scheme and so the obtained accuracy can only give an approximate indication about the final motion compensation accuracy.

6.4 Robust Real-time Visual Measurement

Usually, when doing visual servoing *in vivo*, vision is Achilles' heel. Indeed, endoscopic images are especially difficult to process reliably. This is mainly due to specular highlights caused by the reflection of intense endoscopic light (an annular light source which is around the optical axis at the tip of the endoscope) on the wet surface of the organs. Furthermore, respiration or heart beating can cause large displacements between consecutive images thus increasing the tracking difficulty. Additionally, some organs are very poor in term of landmarks (*e.g.* the surface of the liver which has an uniform texture) and tracking algorithms have difficulties to "stick" to a specified patch. Finally, in a surgery context, the scene is supposed to change during time and occlusions can often occur due to the motion of instruments in front of the endoscope.

To meet the high safety standards required in surgery, the visual feedback must tend to be perfectly reliable. The easiest way to improve this point is to add artificial markers to the scene. So, Nakamura and Ginhoux [31, 19] affixed a target directly onto the epicardium. The main drawback with this approach is the additional time needed to attach the markers: the use of a robotic system should always simplify procedures from the surgeon's point of view. So people started to work on markerless techniques. In [34], Ortmaier proposes a region-based tracking technique using natural landmarks (a set of small patches in the image) with compensation of illumination variations and removal of specular highlights. Takata worked on a condensation algorithm [40]. Noce *et al.* worked on a texture-based tracking algorithm [32]. A texture distance is defined and used to track patches in a high-speed video sequence. The proposed algorithm is compared to the efficient second order minimization (ESM) method proposed by Benhimane and Malis [7] which behaves also very well.

In the context of stereo endoscopy, Stoyanov *et al.* [38] propose a feature-based method that uses a combination of maximally stable extremal regions (MSER) and traditional gradient-based image features. MSER-landmarks detection is robust to illumination variations and specular highlights. The heart is a highly deformable organ. Thus, in [32], the authors use thin-plate splines to parameterize the surface

deformation. They extend the parametrization to the third dimension using stereo endoscopy. A computationally efficient technique is used to compensate for illumination variations.

In the remainder of this section, we will detail one of the most efficient algorithm for heart surface tracking, the ESM method. This algorithm searches the object position and orientation that best matches the object view. It is robust against illumination variation, partial occlusion and specular highlights. Also it is natively robust against small deformations of the object. Furthermore, it can be easily extended to account explicitly for deformations. And, last but not least, it can be implemented to run in real-time at a high sampling rate.

6.4.1 Problem Formulation

Suppose that the object we want to track is a plane and has sufficiently rich texture on the surface. Let the image of the object we capture be I, and the brightness of the point $p = (u, v, 1)$ in the image be $I(p)$. Then the pattern we want to track is expressed by $I^*(p^*)$ where $p^* = (u^*, v^*, 1)$ and $(u^*, v^*) \in \{1, \ldots n\} \times \{1, \ldots m\}$ is the index to identify the point in the tracking area. To simplify the expression we write the point as p_i^* for $i = 1, \ldots, q$ and $q = mn$. The ordering can be either row-major or column-major. Then the brightness of the point is $I^*(p_i^*)$.

The problem can be formalized as a nonlinear estimation problem to find the optimal displacement G that transforms the point p_i^* to the point p_i in the current image so that the brightness of the template and the current image becomes the same:

$$I(p_i) = I^*(p_i^*) \tag{6.9}$$

where $p_i = w(Gp_i^*)$ and with w the perspective transformation.

If the target is planar the transformation is expressed by an homography matrix G. Suppose that there are two camera positions and two images taken from these two positions. The matrix G and the perspective transformation w transforms the plane texture from one viewpoint to the other viewpoint:

$$p_i = w(Gp_i^*) \tag{6.10}$$

where

$$G = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix}, \quad w(Gp_i^*) = \begin{bmatrix} \frac{g_{11}u^* + g_{12}v^* + g_{13}}{g_{31}u^* + g_{32}v^* + g_{33}} \\ \frac{g_{21}u^* + g_{22}v^* + g_{23}}{g_{31}u^* + g_{32}v^* + g_{33}} \\ 1 \end{bmatrix}. \tag{6.11}$$

Estimation of the homography matrix becomes a nonlinear least-square minimization problem: find the matrix G that minimizes the sum of square difference (SSD) error defined by

$$\sum_{i=1}^{q} \left(I(w(G)p_i^*) - I^*(p_i^*)\right)^2. \tag{6.12}$$

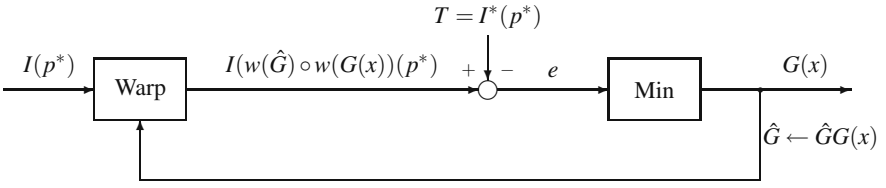


Fig. 6.7 ESM framework

6.4.2 Algorithm

The problem is an old but an important subject in computer vision. Many algorithms to solve this have already proposed [5, 39, 22, 24, 25]. Here we used the Lie algebra to express the homography matrix [7]. This parametrization is useful because it has not singular points.

Let the parametrization of G be

$$G(x) = exp(A(x)) \quad \text{where} \quad A(x) = \sum_{i=1}^{8} x_i A_i \tag{6.13}$$

and A_i $(i = 1, \ldots, 8)$ are a set of base matrices of $sl(3)$ [29].

Based on this parametrization the minimization function is

$$f(x) = y^T(x)y(x) \tag{6.14}$$

where $y(x)$ is q-dimensional vector obtained by stacking the pixel brightness difference

$$y(x) = I(w(G(x)p^*)) - I^*(p^*). \tag{6.15}$$

The minimization algorithm can be a gradient descent, Gauss–Newton or Levenberg–Marquardt method. We proposed an efficient second-order minimization algorithm [7]. It approximates the Hessian matrix with small number of computation. Note that we can have a Taylor series expression of $y(x)$ as

$$y(x) = y(0) + \frac{1}{2}(J(0) + J(x))x + O(x^3). \tag{6.16}$$

Then the approximation at $x = x_0$ is

$$y(x_0) = y(0) + \frac{1}{2}(J(0) + J(x_0))x_0. \tag{6.17}$$

Define the matrix

$$J_{esm} = \frac{1}{2}(J(0) + J(x_0)) \tag{6.18}$$

and the minimization algorithm will be

$$x_0 = J_{esm}^{+} y(0) \tag{6.19}$$

where J_{esm}^{+} is the pseudo-inverse of J_{esm}. The framework of the ESM algorithm is shown in Figure 6.7. The matrix $\hat{G}$ is the best estimation at the previous iteration.

6.4.3 High-speed Implementation

We implemented the ESM algorithm on a dual multi-core processors PC using OpenCV [30], ATLAS [1], and Intel®[2] integrated performance primitives [23]. The algorithm is carefully analyzed and tuned using Google™[3] performance tools [20].

From our analysis the most time consuming part in the algorithm is the computation of the Jacobian matrix J_{esm} and its pseudo-inverse because of the large size of the Jacobian. For example if the size of the tracking patch is 100×100, then the size of the Jacobian is 10000×8. The computation of the Jacobian matrix includes the evaluation of the pixel positions, the image brightness of the corresponding position, the horizontal and vertical gradient of the image brightness and their multiples. So once the camera image comes to the main memory of the host computer, then the computation can be done in parallel. Also the ATLAS package provides an effective parallel thread implementation for computing matrix multiplication. Thus the most important thing is the data dependency in the computation of the Jacobian matrix.

The secondary time consuming task is image display, especially when we use OpenCV. It is necessary for users to check the tracking validity. A 30 fps framerate is however enough to check any error.

Finally, since the tasks for camera capture, image processing and image display are multi-threaded, we have to be careful that these tasks are actually accessing the freshest frame. For our typical environment, the camera capture is 1000 fps, the image processing is 150 fps, and the image display is 30 fps. Thus we adopted a query-answer approach. The display task queries the freshest frame to the processing task, and the processing task queries the freshest frame to the capture (DMA controller) task.

The detail of the implementation is not given here. But with a proper use of the tools the ESM algorithm can run more than 300 fps for a 16×16 pixels template.

6.4.4 Results

The system is composed of a high-speed camera and a multi-core PC. The operating system of the PC is a real-time patched Linux®. The camera is a MC1362 (Camera Link®[4] data link) from Mikrotron®. The frame grabber is a microEnable IV-Fullx4 from Silicon Software ® or an APX-3316 from Aval Data®. This combination achieves 500 fps with 1280×1024 pixels image size and 1000 fps with 640×480 pixels image size.

[2] Intel® is a registered trademark of Intel Corporation in the U.S. and other countries. http://www.intel.com.

[3] Google™ is a trademark owned by Google Inc. www.google.com

[4] Camera Link® is a registered trademark of Mikrotron GmbH. http://www.mikrotron.de.

(a) (b) (c)

Fig. 6.8 The pattern is occluded by a finger or a part of the pattern is high lighted

Fig. 6.9 The book surface is bended and twisted

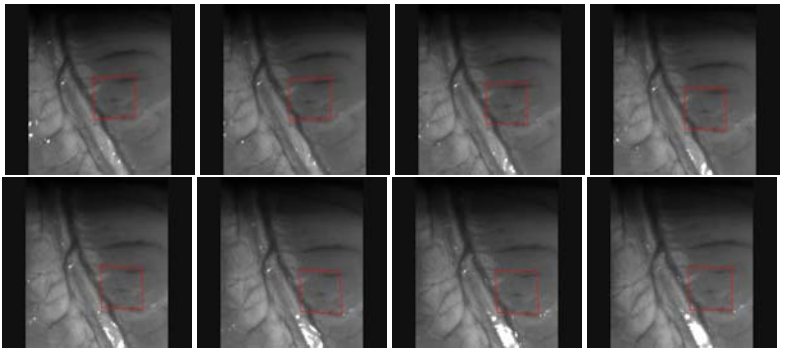


Fig. 6.10 Tracking results

First some preliminary results will be shown. The first results show the robustness of the algorithm with respect to partial occlusion and to illumination changes in the template. The selected template (Figure 6.8(a)) is first occluded by a finger (Figure 6.8(b)), and then enlightened by an additional torch (Figure 6.8(c)). We remark that the tracking of the template is achieved correctly even when the above disturbances are applied. The tracking area is 64×64 pixels and the image processing is 120 fps.

The second preliminary results show the robustness against the object shape change. The algorithm assumes that the object is planar. However it can be tracked even if the surface is slightly bent or twisted. Figure 6.9 shows the results. The tracking area is 144×128 pixels and the image processing is 30 fps. If the deformation

is very large then the original ESM algorithm does not work. Nevertheless, results on the tracking of a deformable object [29] could then be integrated.

The final results we obtained using a heart beating video are shown in Figure 6.10. The video is taken at 500 fps. The ESM algorithm runs currently at 100 fps, which is close to the estimated needed framerate for beating heart surgery.

6.5 Control Strategies

In order to achieve high accuracy tasks, robots have to be controlled using exteroceptive measurements. In robotized cardiac surgery, visual sensors are generally used because they do not introduce any additional cumber to the operating field: surgeons already use cameras to have a visual feedback, especially in totally endoscopic surgery. In order to compensate for the heart motion, high speed vision has to be used [31]. This allows to avoid aliasing when acquiring the fast transients of the heart motion and also to model correctly the robot dynamics.

For safety reasons, medical robots are lightweight and composed of thin arms. These mechanical characteristics induce a low bandwidth of the system which is incompatible with the dynamics required for the heart motion tracking. In order to cope with these limitations predictive control was introduced for heart motion compensation to enlarge the tracking bandwidth [19, 6, 44]. Predictive control laws include prediction algorithms that anticipate future heart motion as described in Section 6.3. Predictive control is more suitable when it is possible to directly measure the current heart motion, *i.e.*, in an active compensation approach. In an active stabilization context, the end-effector of the robotic system is in contact with the heart surface making the direct measurement of the heart motion impossible. In this case, $\mathscr{H}_\infty$ control was suggested [3] in order to tune the robustness of the controlled system with respect to the model uncertainties induced by the contact with the heart muscle.

In this section, the control strategies of two heart motion compensation schemes are described. First, a generalized predictive controller (GPC) implementation for the active compensation approach is described. Then, an $\mathscr{H}_\infty$ control law used for the active stabilization method is derived.

6.5.1 Heart-tool Synchronization

In this approach, a robotic arm is controlled to simultaneously track the motion of the area of interest and move the tool according to the surgeon gesture thanks to a master device.

6.5.1.1 Robotic System Description and Modeling

The testbed (Figure 6.11) consists of a SCARA like robotic arm equipped with a wrist. This robot, holding an instrument and simulating a surgical robot, is actuated

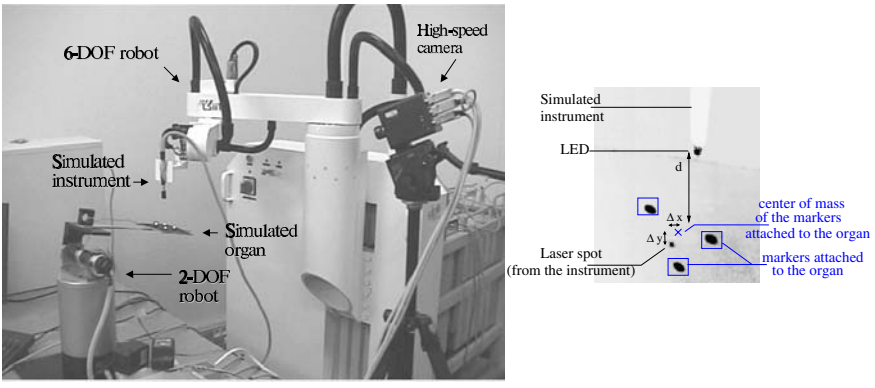


Fig. 6.11 Heart-tool synchronization experimental testbed

using brushless motors with Harmonic Drive®[5] gear-heads. The actuators have been chosen such that their power is high enough to be compatible with the heart motion dynamics. In order to validate the heart motion compensation approach, only the three first joints of the robots are used.

The position of the robot tip with respect to the beating heart is measured thanks to a high speed camera acquiring artificial markers at 500 fps. Indeed four light emitting diodes (LEDs) are used: three are attached to the heart surface while another one is affixed on the instrument tip. Moreover, the instrument embeds a laser source which projects a beam parallel to the instrument axis, yielding a spot on the beating heart. As shown in Figure 6.11, we can define a vector of visual features $F = [d\ \Delta_x\ \Delta_y]$ that fully describes the relative position between the heart and the surgical instrument. Servoing $F = [d\ \Delta_x\ \Delta_y]^T$ to $F^* = [d^*\ 0\ 0]$ yields a synchronization between the heart and the robot motion and a constant relative position between the instrument and the operation area. The visual servo-loop, synchronized with image acquisition, is implemented in a real-time mode on a vision computer hosting the frame grabber. The vision computer performs the image acquisition, image processing and control signals computing. The computed control signals are sent to the robot controller via a 10 Mb/s serial link.

The visual servoing loop is given in Figure 6.12. The computed control law $U(z)$ is delayed by one sample to model the fast serial link transfer, and then converted into an analog voltage $\dot{Q}^*$ with a digital-analog converter modeled by a zero-order hold (ZOH). $\dot{Q}^*$ is then sent as a reference to the joint velocity loops of the robot. The dynamics of these loops are assumed to be almost linear around the current working point and so are modeled by a transfer matrix $G(s)$. The relationship between the current joint velocity vector $\dot{Q}(s)$ and the image velocity vector $\dot{F}(s)$ is given by J_i, the interaction matrix. The integrator between $\dot{F}(s)$ and $F(s)$ models

[5] Harmonic Drive® is a registered trademark of Harmonic Drive LLC. http://www.harmonicdrive.net

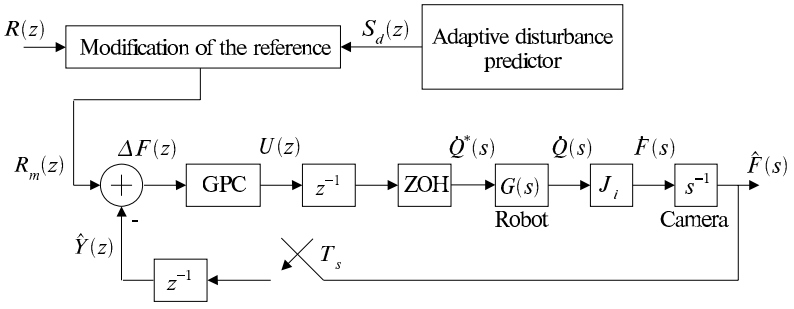


Fig. 6.12 Heart-tool synchronization: a block diagram of the controlled plant

the action of the camera which is intrinsically a position sensor. The delay z^{-1} in feedback loop is due to the image acquisition and processing duration.

The discrete transfer function describing the linearized behavior of the open visual loop around a working point can thus be derived as

$$V(z) = \frac{Y(z)}{U(z)} = z^{-2}(1-z^{-1})\mathscr{Z}\left\{\frac{J_i G(s)}{s^2}\right\}, \tag{6.20}$$

where $\mathscr{Z}$ is the z-transform. In practice, $V(z)$ is identified using standard linear identification techniques.

6.5.1.2 Control Strategy: GPC Controller

In order to design an unconstrained GPC [9], the system model is first converted to an autoregressive and integrated moving average with exogenous input (ARIMAX) equation:

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta(q^{-1})}\xi(t) \tag{6.21}$$

where q^{-1} is the backward operator, A and B are two polynomials modeling the system dynamics (B may also include pure delays), and polynomial C is used to color the zero-mean white noise ξ(t). Polynomial Δ is used to make ξ/Δ a non-stationary noise , which is suitable to model any perturbation in a control loop. For instance, Δ is set to a pure integrator, $\Delta(q^{-1}) = (1-q^{-1})$ when disturbances are only supposed to be constant steps.

The GPC algorithm consists of computing a control sequence that minimizes the cost function

$$\mathscr{J}(u,k) = \sum_{j=N_1}^{N_2} \|\hat{y}(k+j) - r(k+j)\|^2 + \lambda \sum_{j=1}^{N_3} \|\delta u(k+j-1)\|^2$$

where $\hat{y}(k+j)$ is a j-step ahead prediction of the system output, N_1 and N_2 are the lower and upper bounds of the cost horizon and N_3 is the control cost horizon. λ weights the control energy.

The visual measurement variation is due to the heart motion and also to the robot displacements. Thanks to the eye-to-hand configuration of the system, one can easily extract the heart motion S_d from the visual measurement. As explained in Section 6.3, the future heart motion at sample time $k+j$ can be predicted. By modifying the reference of the visual loop (Figure 6.12) as

$$r_m(k+j) = r(k+j) - (S_d(k+j) - S_d(k)), \tag{6.22}$$

the GPC controller can anticipate the future heart motion disturbance yielding a smaller tracking error. A GPC controller with a reference modification is called GPC with adaptive predictor (GPC+A).

6.5.1.3 *In vivo* Results

The control strategy of the robotic tracking of the heart motion was validated during *in vivo* experiments on pigs (Figure 6.13) which underwent full sternotomy after receiving general anesthesia. During the experiments, markers are sticked on the heart surface. The control strategy was validated using a simple GPC (without modifying the reference) and then with the GPC+A (with the modification of the reference). The results show a reduction of 80% of the disturbance with the standard GPC. Reductions of 93% and 84% are obtained respectively for Δ_x and Δ_y when GPC+A is used. These results demonstrate the validity of the approach. Figure 6.14 shows the residual error along Δ_y.

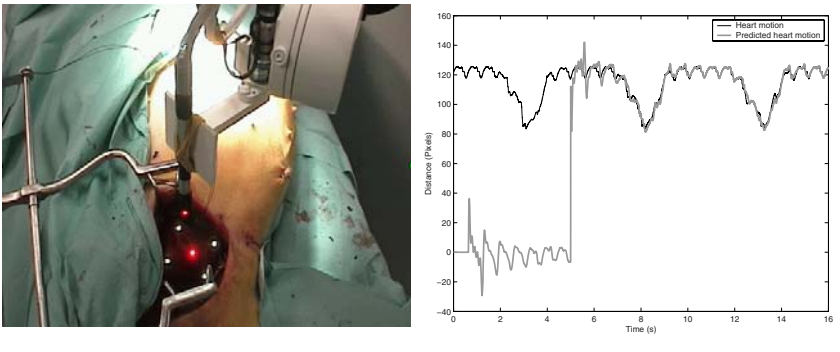


Fig. 6.13 On the left, a picture of an *in vivo* heart-tool synchronization experiment. On the right, the achieved prediction of future heart motion

6.5.2 Heart Immobilization

In this approach, an active stabilizer is controlled to cancel any displacement of the area of interest, while the surgeon is performing the suturing gesture with a telemanipulated robot. We put here the focus on the control of the active stabilizer.

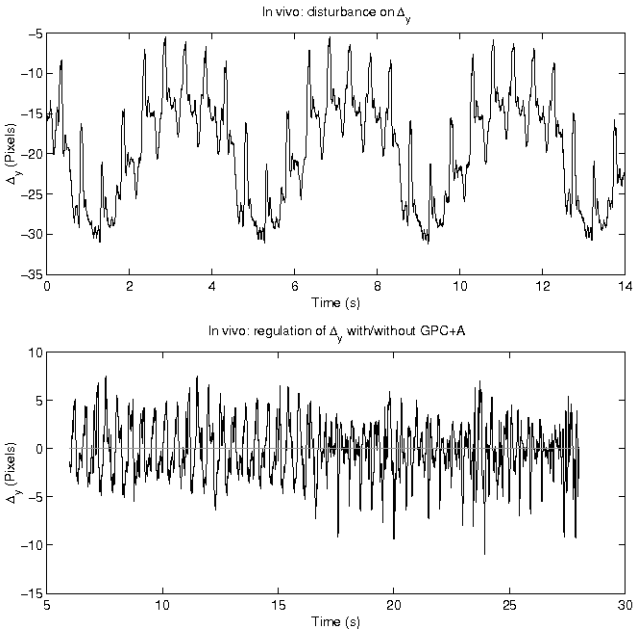


Fig. 6.14 *in vivo* heart-tool synchronization: on the top, the recorded disturbance and on the bottom the tracking error

6.5.2.1 Robotic System Description and Modeling

The Cardiolock active cardiac stabilizer described in Section 6.2 is used. Figure 6.15 shows a block diagram of the visual loop. The computed control signal u is converted into an analog voltage with a digital to analog conversion modeled by a ZOH. This voltage corresponds to the reference of the piezo position servo loop α^*. Due to the high dynamics of this loop, we assume that $\alpha^* = \alpha$. The output y is measured using the position of a visual marker v in the image given by a high speed camera with a sampling period $T_s = 3$ ms. The dynamic effects of the camera can be modeled as an averaging filter representing the exposure time effect. The camera model C_m can be written [36] as $\mathrm{C}_m(z) = \frac{1+z^{-1}}{2}$.

A time delay of one period is required for the acquisition and the processing of the current image. The whole open loop model of the visual servoing can be written as

$$\mathrm{H}(z) = (1-z^{-1})\mathscr{Z}\{\frac{G(s)}{s}\}\mathrm{C}_m(z)z^{-1}. \tag{6.23}$$

This model is identified using standard identification techniques. P representing the transfer between the heart force applied on the stabilizer and the stabilizer tip position can be easily obtained from $\mathrm{H}(z)$.

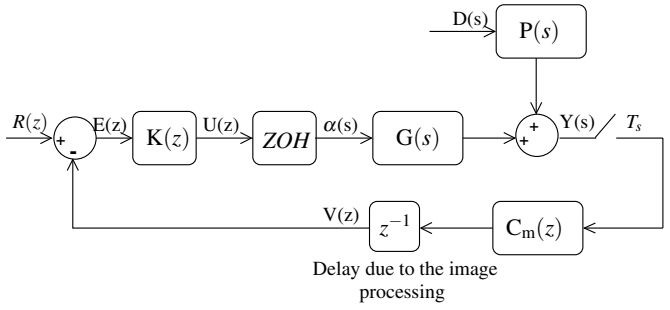


Fig. 6.15 Heart immobilization: a block diagram of the controlled plant

6.5.2.2 Control Strategy: $\mathscr{H}_\infty$ Controller

A feedback control law is designed in order to reject the perturbations due to the heart motion, *i.e.* to keep the stabilizer end effector in a steady position. The control synthesis scheme is given in Figure 6.16. G_c represents the bilinear transform of H, and P_c the bilinear transform of the transfer between the heart force and the visual measurement. The synthesis scheme relies on three weighting functions W_1, W_2 and W_3. The controller $K(s)$ is designed [15] in order to stabilize the closed loop and minimize the $\mathscr{H}_\infty$ norm of the performance channel (the transfer between $w = [w_1, w_2]$ and $z = [z_1, z_2]$). Denoting γ the obtained performance index, we obtain the inequalities

$$\begin{aligned} |T_{er}(j\omega)| &< \gamma |W_1^{-1}(j\omega)| \\ |T_{ur}(j\omega)| &< \gamma |W_3^{-1}(j\omega)| \\ |T_{ed}(j\omega)| &< \gamma |W_1^{-1}(j\omega) W_2^{-1}(j\omega)| \\ |T_{ud}(j\omega)| &< \gamma |W_3^{-1}(j\omega) W_2^{-1}(j\omega)|. \end{aligned} \tag{6.24}$$

The second terms of these inequalities are considered as templates for the transfer function of the left-hand side terms and have to be tuned according to the control objectives. Therefore W_1, W_2 and W_3 have to be chosen according to the control objectives. W_1 is used to set the required modulus margin in order to guarantee a good robustness with respect to the model uncertainties. W_3 is chosen such that the controller gain decreases in the high frequencies in order to be robust with respect to the system unmodeled dynamics. W_2 is set to limit the gain of T_{ed} in the frequency range of interest, *i.e.* where the frequency content of the cardiac perturbation lies.

6.5.2.3 *In vivo* Results

Figure 6.17 reports the results of an *in vivo* stabilization test. The controller was switched on 6 s after the beginning of the experiment. The peak to peak heart excursion was then divided by 4. The RMS of the residual motion is 0.37 mm and 0.03 mm, respectively before and after the activation of the controller.

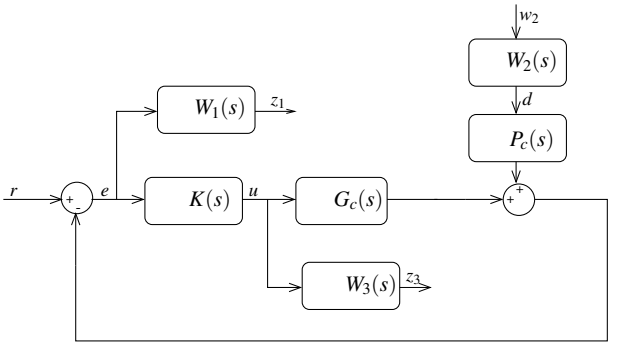


Fig. 6.16 Augmented plant for $\mathscr{H}_\infty$ control synthesis

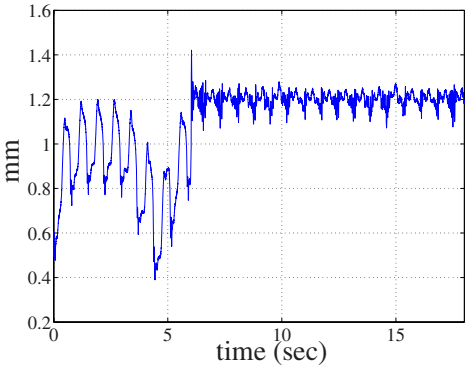


Fig. 6.17 Heart immobilization: *in vivo* results

In the heart-tool synchronization approach, taking into account the prior knowledge of the heart motion has been proved to enhance the motion compensation accuracy. Even if the results obtained by the active stabilizer are promising, we think that they can be improved if a predictive control approach is used. As argued in the beginning of this section, the disturbance induced by the heart motion can not be directly measured, so future research could investigate the ways of estimating this disturbance in order to make the use of predictive control possible.

6.6 Perspectives

An interesting alternative to CABG is angioplasty performed during percutaneous coronary intervention (PCI). The goal is usually to bring the tip of a catheter at the stenosis site using canulation techniques with femoral artery access. The inflation of a balloon or even the implantation of a tubular mesh (a stent) can help expanding the artery wall in order to reduce the stenosis. So many teams are currently working on the development of active catheters able to navigate more easily through the

vessels' ramifications. Visual servoing using medical imaging feedback (MRI, CT, fluoroscopy) could be used to automatically guide the tip of the catheter towards the stenosis. However, PCI is only suitable for non-severe coronary artery disease. Therefore, CABG is still the only solution for many patients and probably for a long time.

Off-pump TECAB is surely the future of CABG. Visual servoing can help simplifying the procedure for the surgeon mainly by improving the stabilization of the suturing site. It is possible that additional improvements are needed to achieve this complicated gesture with maximal safety. For example, the use of redundant modalities to track the coronary artery could improve the robustness and the quality of the task (*e.g.* by using ultrasound imaging). More generally, visual servoing using medical imaging feedback seems to be a promising way of research. A great boost could be given in this direction by improving the real-time low-level access to the image data through a new dedicated normalized channel. But almost everything has still to be done on this matter. Only tight collaborations between medical imaging device manufacturers and visual servoing experts could yield such a new standard.

References

[1] ATLAS: Atlas, `http://math-atlas.sourceforge.net/`
[2] Bachta, W., Renaud, P., Cuvillon, L., Laroche, E., Forgione, A., Gangloff, J.: Motion prediction for computer-assisted beating heart surgery. IEEE Transactions on Biomedical Engineering (to appear)
[3] Bachta, W., Renaud, P., Laroche, E., Forgione, A., Gangloff, J.: Cardiolock: an active cardiac stabilizer, first in vivo experiments using a new robotized device. Computer Aided Surgery 13(5), 243–254 (2008)
[4] Bachta, W., Renaud, P., Laroche, E., Gangloff, J.: Cardiolock2: Parallel singularities for the design of an active heart stabilizer. In: IEEE International Conference on Robotics and Automation (2009)
[5] Baker, S., Matthews, I.: Lucas-kanade 20 years on: A unifying framework. Int. J. of Computer Vision 56(3) (2004)
[6] Bebek, O., Cavusoglu, M.: Intelligent control algorithms for robotic-assisted beating heart surgery. IEEE Transactions on Robotics 23(3), 468–480 (2007)
[7] Benhimane, S., Malis, E.: Homography-based 2D visual tracking and servoing. Int. J. Robotics Research 26(7), 661–676 (2007)
[8] Boyd, W.D., Desai, N.D., Rizzo, D.F.D., Novick, R.J., McKenzie, F.N., Menkis, A.H.: Off-pump surgery decreases post-operative complications and resource utilization in the elderly. Annals of Thoracic Surgery 68, 1490–1493 (1999)
[9] Camacho, E.F., Bordons, C.: Model Predictive Control. Springer, London (1999)
[10] Cattin, P., Dave, H., Grunenfelder, J., Szekely, G., Turina, M., Zund, G.: Trajectory of coronary motion and its significance in robotic motion cancellation. European Journal of Cardio-thoracic Surgery 25(5), 786–790 (2004)
[11] Choi, D., Riviere, C.: Flexure-based manipulator for active handhled microsurgical instrument. In: 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (2005)

[12] Crick, S.J., Sheppard, M.N., Ho, S.Y., Gebstein, L., Anderson, R.H.: Anatomy of the pig heart: comparisons with normal human cardiac structure. Journal of Anatomy 193(1), 105 (1998)
[13] Cuvillon, L., Gangloff, J., De Mathelin, M., Forgione, A.: Towards robotized beating heart TECABG: assessment of the heart dynamics using high-speed vision. Computer aided surgery 11(5), 267–277 (2006)
[14] Dario, P., Hannaford, B., Menciassi, A.: Smart surgical tools and augmenting devices. IEEE Transactions on Robotics and Automation 19(5), 782–792 (2003)
[15] Doyle, J., Glover, K., Khargonekar, P., Francis, B.: State-space solutions to standard H_2 and H_∞ control problems. IEEE Transactions on Automatic Control 34(8), 831–847 (1989)
[16] Duchemin, G., Poignet, P., Dombre, E., Pierrot, F.: Medically safe and sound [human-friendly robot dependability]. IEEE Robotics and Automation Magazine 11(2), 46–55 (2004)
[17] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation 8(3), 313–326 (1992)
[18] Falk, V., Walther, T., Stein, H., Jacobs, S., Walther, C., Rastan, A., Wimmer-Greinecker, G., Mohr, F.: Facilated endoscopic beating heart coronary artery bypass grafting using a magnetic coupling device. The Journal of Thoracic and Cardiovascular Surgery 126(5), 1575–1579 (2003)
[19] Ginhoux, R., Gangloff, J., de Mathelin, M., Soler, L., Sanchez, M., Marescaux, J.: Active filtering of physiological motion in robotized surgery using predictive control. IEEE Transactions on Robotics 21(1), 67–79 (2005)
[20] Google-perftools, `http://code.google.com/p/google-perftools/`
[21] Gummert, J., Opfermann, U., Jacobs, S., Walther, T., Kempfert, J., Mohr, F., Falk, V.: Anastomotic devices for coronary artery bypass grafting: technological options and potential pitfalls. Computers in Biology and Medicine 37, 1384–1393 (2007)
[22] Hager, G., Belhumeur, P.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(10), 1125–1139 (1998)
[23] IPP: Ipp, `http://software.intel.com/en-us/intel-ipp/`
[24] Jin, H., Favaro, P., Soatto, S.: Real-time feature tracking and outlier rejection with changes in illumination. In: Proc. IEEE Int. Conf. on Computer Vision, vol. 1, pp. 684–689 (2001)
[25] Jurie, F., Dhome, M.: Hyperplane approximation for template matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 996–1000 (2002)
[26] Kettler, D., Plowes, R., Novotny, P., Vasilyev, N., del Nido, P., Howe, R.: An active motion compensation instrument for beating heart mitral valve surgery. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2007. IROS 2007, pp. 1290–1295 (2007)
[27] Lemma, M., Mangini, A., Redaelli, A., Acocella, F.: Do cardiac stabilizers really stabilize? experimental quantitative analysis of mechanical stabilization. Interactive Cardiovascular and Thoracic Surgery (2005)
[28] Loisance, D., Nakashima, K., Kirsch, M.: Computer-assisted coronary surgery: lessons from an initial experience. Interactive Cardiovascular and Thoracic Surgery 4, 398–401 (2005)
[29] Malis, E.: An efficient unified approach to direct visual tracking of rigid and deformable surfaces. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2729–2734 (2007)
[30] OpenCV: `http://sourceforge.net/projects/opencvlibrary/`

[31] Nakamura, Y., Kishi, K., Kawakami, H.: Heartbeat synchronization for robotic cardiac surgery. In: IEEE International Conference on Robotics and Automation, vol. 2, pp. 2014–2019 (2001)
[32] Noce, A., Triboulet, J., Poignet., P.: Efficient tracking of the heart using texture. In: Proceedings of IEEE International Conference of the Engineering in Medicine and Biology Society (EMBS 2007), Lyon, France, vol. 1, pp. 4480–4483 (2007)
[33] OECD: Health data: Statistics and indicators for 30 countries, `http://www.oecd.org/health/healthdata`
[34] Ortmaier, T.: Motion compensation in minimally invasive robotic surgery. Ph.D. thesis, Technischen Universität München (2002)
[35] Patronik, N., Zenati, M., Riviere, C.: Preliminary evaluation of a mobile robotic device for navigation and intervention on the beating heart. Computer Aided Surgery 10(5), 225–232 (2005)
[36] Ranftl, A., Cuvillon, L., Gangloff, J., Sloten, J.: High speed visual servoing with ultrasonic motors. In: IEEE Int. Conf. on Robotics and Automation, pp. 4472–4477 (2007)
[37] Salcudean, S., Lichtenstein, S., Trejos, A., Sassani F.and Gihuly, T.: Moving tracking platform for relative motion cancellation for surgery. US Patent 6368332B1, April 9 (2002)
[38] Stoyanov, D., Mylonas, G.P., Deligianni, F., Darzi, A.: Soft-tissue motion tracking and structure estimation for robotic assisted MIS procedures. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3750, pp. 139–146. Springer, Heidelberg (2005)
[39] Sugimoto, S., Okutomi, M.: A direct and efficient method for piecewise-planar surface reconstruction from stereo images. In: Proc. IEEE Society Conf. on Comptuer Vision and Pattern Recognition, pp. 1–8 (2007)
[40] Takata, U.: Thesis. Ph.D. thesis, Graduate School of Frontier Sciences, The University of Tokyo (2009)
[41] Thakral, A., Wallace, J., Tolmin, D., Seth, N., Thakor, N.: Surgical motion adaptive robotic technology (S.M.A.R.T): Taking the motion out of physiological motion. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 317–325. Springer, Heidelberg (2001)
[42] Trejos, A., Salcudean, S., Sassani, F., Lichtenstein, S.: On the feasibility of a moving support for surgery on the beating heart. In: Taylor, C., Colchester, A. (eds.) MICCAI 1999. LNCS, vol. 1679, pp. 1088–1097. Springer, Heidelberg (1999)
[43] Van Dijk, D., Jansen, E., Hijman, R.E.A.: Cognitive outcome after off-pump and on-pump coronary artery bypass graft surgery. Journal of the American Medical Association 287(11), 1405–1412 (2002)
[44] Yuen, S., Novotny, P., Howe, R.: Quasiperiodic predictive filtering for robot-assisted beating heart surgery. In: IEEE International Conference on Robotics and Automation, pp. 3875–3880 (2008)

Chapter 7
A Variational Approach to Trajectory Planning in Visual Servoing

Youcef Mezouar

Abstract. One deficiency of image-based servo is that the induced 3D trajectories are not optimal and sometimes, especially when the displacements to realize is large, these trajectories are not physically valid leading to the failure of the servoing process. In this chapter, we address the problem of generating trajectories of some image features that corresponds to optimal 3D trajectories in order to control efficiently a robotic system using an image-based control strategy. First, a collineation path between given start and end points is obtained and then the trajectories of the image features are derived. Path planning is formulate as a variational problem which allows to consider simultaneously optimality and inequality constraints (visibility). A numerical method is employed for solving the path-planning problem in the variational form.

7.1 Introduction

Image-based servoing is now a well-known local control framework [2, 3]. In this approach, the reference image of the object corresponding to a desired position of the robot is acquired first (during an off-line step) and some image features are extracted. Features extracted from the initial image are matched with those obtained from the desired one. These features are then tracked during the robot (and/or the object) motion, using for example a correlation based method. An error is obtained by comparing the image features in the current image and in the reference one. The robot motion is then controlled in order to minimize the error (using for example a gradient descent approach). Since the error is directly measured in the image, image-based servo has some degrees of robustness with respect to modeling errors and noise perturbations [24, 1]. However, sometimes, and especially when the initial and desired configurations are distant, the trajectories induced by image-based servo

Youcef Mezouar
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France
e-mail: mezouar@univ-bpclermont.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 117–132.
springerlink.com

are neither physically valid nor optimal due to the nonlinearity and singularities in the relation from the image space to the workspace [1].

Dealing with this deficiency, an approach consists on selecting features with good decoupling and linearizing properties. In [18], vanishing points have been used for a dedicated object (a 3D rectangle), to obtain some decoupling properties. For the same object, six visual features have been designed in [7] to control the 6 degrees of freedom (DOF) of a robot arm, following a partitioned approach. In [17], the coordinates of points are expressed in a cylindrical coordinate system instead of the classical Cartesian one, so as to improve the robot trajectory. In [14], the three coordinates of the centroid of an object in a virtual image obtained through a spherical projection have been selected to control 3 DOF of an under-actuated system. In [19], Mahony *et al.* deals with the selection of the optimal feature to control the camera motion with respect to the depth axis. Tatsambon *et al.* in [12] proposed a decoupled visual servoing from spheres using a spherical projection model. In [9], translational and rotational motions are decoupled by using the homography and the epipolar constraint between the current and the goal images. In [27], moments in the unit sphere allow to design partitioned systems with good decoupling and linearizing properties.

An other approach consists on coupling path-planning method to image-based servoing. Indeed, if the initial error is too large, a reference trajectory can be designed from a sequence of images. The initial error can thus be sampled so that, at each iteration of the control loop, the error to regulate remains small. In [15], relay images that interpolate initial and reference image features using an affine approximation of the relationship between initial and desired images, coupled to a potential switching control scheme, is proposed to enlarge the stable region. In [16], a trajectory generator using a stereo system is proposed and applied to obstacle avoidance. An alignment task for an 4 DOF robot using intermediate views of the object synthesized by image morphing is presented in [26]. A path-planning for a straight-line robot translation observed by a weakly calibrated stereo system is performed in [25]. In [22], a potential field-based path-planning generator that determines the trajectories in the image of a set of points lying on an unknown target is presented. To increase the stability region, [8] describes a globally stabilizing method using navigation function for eye-to-hand setup while in [6], the authors propose to use circular-like trajectories. In [5], the camera is constrained to follow a straight line while ensuring visibility.

However, none of these works were dealing with optimality issues. In [29], a numerical framework for the design of optimal trajectories in the image space is described and applied to the simple case of a one dimensional camera in a two dimensional workspace. In [23] an analytical solution to none-constrained optimal path-planning in the image space for general setup is presented. In this chapter, we address the problem of generating trajectories of image features that corresponds to optimal 3D trajectories under visibility constraints in order to control efficiently a robotic system using an image-based control strategy. First, a collineation path between given start and end points is obtained and then the trajectories of the image features are derived. Path planning is formulate as a variational problem which

allows to consider simultaneously optimality and inequality constraint (visibility). A numerical method is employed for solving the path-planning problem in the variational form.

7.2 Preliminaries

This section briefly recalls some mathematical backgrounds on the rotational group, the standard perspective projection and the associated two views geometry. It also presents the essential of the path planning method proposed in [23].

7.2.1 Brief Review of $SO(3)$

The group $SO(3)$ is the set of all 3×3 real orthogonal matrices with unit determinant and it has the structure of a Lie group. On a Lie group, the space tangent to the identity has the structure of a Lie algebra. The Lie algebra of $SO(3)$ is denoted by $so(3)$. It consists of the 3×3 skew-symmetric matrices, so that the elements of $so(3)$ are matrices of the form

$$[\theta] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}.$$

One of the main connections between a Lie group and its Lie algebra is the exponential mapping. For every $\mathbf{R}\in SO(3)$, there exists at least one $[\theta]\in so(3)$ such that $e^{[\theta]} = \mathbf{R}$ with (Rodriguez formula)

$$\mathbf{R} = e^{[\theta]} = \mathbf{I} + \frac{\sin\|\theta\|}{\|\theta\|}[\theta] + \frac{1-\cos\|\theta\|}{\|\theta\|^2}[\theta]^2, \tag{7.1}$$

where $\|\theta\|$ is the standard Euclidean norm. Conversely, if $\mathbf{R}\in SO(3)$ such that $\mathrm{Trace}(\mathbf{R}) \neq -1$, then

$$[\theta] = \log(\mathbf{R}) = \frac{\theta}{2\sin\theta}(\mathbf{R}-\mathbf{R}^T), \tag{7.2}$$

where θ satisfies

$$\theta = \|\theta\| = \arccos\left(\frac{\mathrm{Trace}(\mathbf{R})-1}{2}\right). \tag{7.3}$$

If $\mathrm{Trace}(\mathbf{R}) = -1$, $\log(\mathbf{R})$ can be obtained noticing that $\theta = \pm\pi\mathbf{u}$ where $\mathbf{u}$ is a unit length eigenvector of $\mathbf{R}$ associated with the eigenvalue 1.

Another important connection between $so(3)$ and $SO(3)$ involves angular velocities. If $\mathbf{R}(t)$ is a curve in $SO(3)$, then $\dot{\mathbf{R}}\mathbf{R}^T$ and $\mathbf{R}^T\dot{\mathbf{R}}$ are skew-symmetric, and hence element of $so(3)$. The element ω of $so(3)$ such that

$$[\omega] = \mathbf{R}^T\dot{\mathbf{R}} \tag{7.4}$$

corresponds to the angular velocity of the rigid body.

7.2.2 Camera Model and Two Views Geometry

The standard perspective camera model maps all scene points $\mathscr{X}$ with homogeneous coordinates in the camera frame $\mathbf{X} = [X\ Y\ Z\ 1]^T$ from a line passing through the optical center of the camera to one image point with homogeneous coordinates $\mathbf{m} = [m_x\, m_y\, 1]^T$ in the normalized image plane:

$$\lambda \mathbf{m} = \mathbf{P}\mathbf{X}. \tag{7.5}$$

$\mathbf{P} \in \mathbb{R}^{3\times 4}$ is the projection matrix, that is $\mathbf{P} = [\mathbf{I}_{3\times 3}|\mathbf{0}_{3\times 1}]$. The 2D projective point $\mathbf{m}$ is then mapped into the pixel image point with homogeneous coordinates $\mathbf{p} = [x\, y\, 1]^\top$ using the collineation matrix $\mathbf{K}$:

$$\mathbf{p} = \mathbf{K}\mathbf{m} \tag{7.6}$$

where $\mathbf{K}$ contains the intrinsic parameters of the camera:

$$\mathbf{K} = \begin{bmatrix} fp_u & -fp_u \cot(\alpha) & u_0 \\ 0 & fp_v/\sin(\alpha) & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

u_0 and v_0 are the pixels coordinates of principal point, f is the focal length, p_u and p_v are the magnifications respectively in the u and v directions, and α is the angle between these axes.

Consider now two views of a scene observed by a camera (see Figure 7.1). A 3D point $\mathscr{X}$ with homogeneous coordinates $\mathbf{X} = [X\ Y\ Z\ 1]^T$ is projected under perspective projection to a point $\mathbf{p}$ in the first image (with homogeneous coordinates measured in pixel $\mathbf{p} = [x\, y\, 1]^T$) and to a point $\mathbf{p}^f$ in the second image (with homogeneous coordinates measured in pixel $\mathbf{p}^f = [x^f\ y^f\ 1]^T$). It is well-known that there exists a projective homography matrix $\mathbf{G}$ related to a virtual plane Π, such that for all points $\mathscr{X}$ belonging to Π[1],

$$\mathbf{p} \propto \mathbf{G}\mathbf{p}^f.$$

When $\mathbf{p}$ and $\mathbf{p}^f$ are expressed in pixels, matrix $\mathbf{G}$ is called the collineation matrix. From the knowledge of several matched points, lines or contours [10, 21], it is possible to estimate the collineation matrix. For example, if at least four points belonging to Π are matched, $\mathbf{G}$ can be estimated by solving a linear system. Else, at least eight points (3 points to define Π and 5 outside of Π) are necessary to estimate the collineation matrix by using for example the linearized algorithm proposed in [20]. Assuming that the camera calibration is known, the Euclidean homography can be computed up to a scalar factor[2]:

$$\mathbf{H} \propto \mathbf{K}^{+}\mathbf{G}\mathbf{K}. \tag{7.7}$$

[1] $\mathbf{p} \propto \mathbf{G}\mathbf{p}^f \Longleftrightarrow \alpha\mathbf{p} = \mathbf{G}\mathbf{p}^f$ where α is a scaling factor.

[2] $\mathbf{K}^{+}$ denotes the inverse of $\mathbf{K}$.

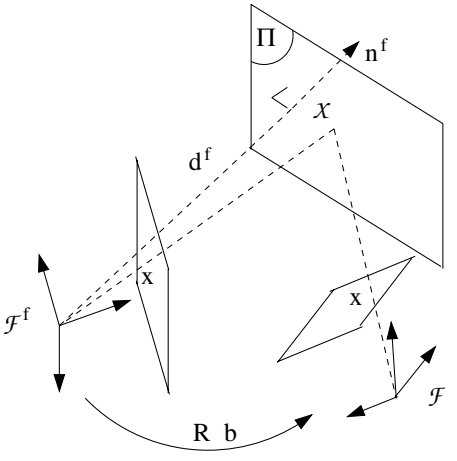


Fig. 7.1 Geometry of two views

The Euclidean homography can be decomposed into a rotation matrix and a rank 1 matrix [11]:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{b}}{d^f}\mathbf{n}^{fT}, \tag{7.8}$$

where $\mathbf{R}$ and $\mathbf{b}$ represent the rotation matrix and the translation vector between the current and the desired camera frames (denoted $\mathcal{F}$ and $\mathcal{F}^f$ respectively), $\mathbf{n}^f$ is the unitary normal to the virtual plane expressed in $\mathcal{F}^f$, and d^f is the distance from Π to the origin of $\mathcal{F}^f$ (see Figure 7.1). From $\mathbf{G}$ and $\mathbf{K}$, it is thus possible to determine the camera motion parameters (*i.e.*, the rotation $\mathbf{R}$ and the scaled translation $\mathbf{b}_{d^f} = \frac{\mathbf{b}}{d^f}$) and the normal vector $\mathbf{n}^f$, by using for example one of the algorithms proposed in [11] or in [30].

In the sequel we assume that given an initial image and a desired image of the scene, some image features can be extracted and matched. This framework is the classical one in visual servoing. From the extracted image features, the collineation matrix at time t_0, $\mathbf{G}_0$, can be computed [10, 4]. Note also that, when the desired configuration is reached (at time t_f), the collineation matrix is proportional to the identity matrix:

$$\mathbf{G}^f \propto \mathbf{I}_{3\times 3}.$$

In the next sections, we consider the problem of finding a path of the collineation matrix between $\mathbf{G}_0$ and $\mathbf{G}^f$ corresponding to an optimal camera path with respect to criteria which will be specified in the sequel. The image trajectories are then derived from the collineation path.

7.2.3 The Unconstrained Problem

The constrained variational problem we will solve in the sequel requires an initializing trajectory. We will use the trajectory provided by the method proposed in [23] by

solving a minimum acceleration problem . Let us recall the essential of this method. We assume that the current position of the camera with respect to its desired position is given by the rotation matrix $\mathbf{R}(t)$ and the translation vector $\mathbf{b}(t)$. In this case, the collineation matrix is given by

$$\mathbf{G}(t) \propto \mathbf{K}^{+} \left(\mathbf{R}(t) + \mathbf{b}_{d^f}(t) \mathbf{n}^{fT} \right) \mathbf{K}.$$

We denote $\mathbf{U}$ the 6×1 vector $[\mathbf{v}^T \omega^T]^T$, where $\mathbf{v}$ denotes the time derivative of $\mathbf{b}$ and ω is defined by (7.4). In [23], the problems, denoted **PC1** and **PC2**, of finding a path of the collineation matrix corresponding to the minimum energy and minimum acceleration problem respectively have been solved. These problems can be formulated as

- *(PC1)* find $\mathbf{G}(t)$ minimizing:

$$J_1 = \int_0^1 \mathbf{U}^T \mathbf{U} dt,$$

  subject to (7.4), $\mathbf{v} = \dot{\mathbf{b}}$ and with boundary conditions:

$$\begin{cases} \mathbf{G}(0) \propto \mathbf{G}_0, \\ \mathbf{G}(1) \propto \mathbf{I}_{3\times 3}; \end{cases}$$

- *(PC2)* find $\mathbf{G}(t)$ minimizing:

$$J_2 = \int_0^1 \dot{\mathbf{U}}^T \dot{\mathbf{U}} dt,$$

  subject to (7.4), $\mathbf{v} = \dot{\mathbf{b}}$ and with boundary conditions:

$$\begin{cases} \mathbf{G}(0) \propto \mathbf{G}_0, \\ \mathbf{G}(1) \propto \mathbf{I}_{3\times 3}, \\ \mathbf{U}(0) = \mathbf{0}_{6\times 1}, \\ \mathbf{U}(1) = \mathbf{0}_{6\times 1}. \end{cases}$$

In this case, the camera velocity is constrained to be $\mathbf{0}$ at the beginning and the end of the task. The boundary conditions are verified if $\mathbf{R}(0) = \mathbf{R}_0$, $\mathbf{b}(0) = \mathbf{b}_0$, $\mathbf{R}(1) = \mathbf{I}_{3\times 3}$ and $\mathbf{b}(1) = \mathbf{0}$. The solutions of **PC1** and **PC2** are given by the following proposition [23].

Proposition 7.1. *The optimal path of the collineation matrix in the sense of* **PC1** *and* **PC2** *is given by*

$$\mathbf{G}(t) \propto (1 - q(t)) \Phi_0 + (\mathbf{G}_0 - \Phi_0) \Gamma(\theta_0, t) \tag{7.9}$$

where

$$\Gamma(\theta_0, q(t)) = \mathbf{K}\mathbf{e}^{[\theta_0]q(t)}\mathbf{K}^+ \text{ and } \Phi_0 = \mathbf{K}\mathbf{b}_{0d^f}\mathbf{n}^{fT}\mathbf{K}^+ \tag{7.10}$$

with $[\theta_0] = \log(\mathbf{R}_0^T)$, $\mathbf{b}_{0d^f} = \frac{\mathbf{b}_0}{d^f}$ *and*

$$\begin{cases} q(t) = t & \text{if } \mathbf{PC1}, \\ q(t) = -2t^3 + 3t^2 & \text{if } \mathbf{PC2}. \end{cases}$$

The path given by Proposition 7.1 corresponds to a shortest distance path of the rotation matrix (minimal geodesic) with respect to an adequately chosen Riemannian metric on $SO(3)$ and to a straight line translation.

7.3 The Constrained Problem

As previously, we assume that the current position of the camera with respect to its desired position is given by the rotation matrix $\mathbf{R}(t)$ and the translation vector $\mathbf{b}(t)$. Let $\mathbf{u}$ and θ be the axis and the rotation angle obtained from $\mathbf{R}(t)$ and define the camera state as $\mathbf{x} = [\mathbf{u}\theta\ \mathbf{b}_{d^f}]^\top$. Consider now the dynamical system described by the state equations

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{U}(t), t) = \mathbf{U}. \tag{7.11}$$

$\mathbf{U}$ denotes the input vector. The state is known at the initial $(t = 0)$ and final time $(t = t_f)$. The problem is to find a piecewise smooth input vector $\mathbf{U}$ from a known initial state $\mathbf{x}_0$ to a desired $\mathbf{x}_f$ so that the cost function

$$J = \int_{t_0}^{t_f} \mathbf{U}^T\mathbf{U}dt \tag{7.12}$$

is minimized with boundary conditions

$$\mathbf{G}(t_0) \propto \mathbf{G}_0,$$

$$\mathbf{G}(t_f) \propto \mathbf{I}_{3\times 3},$$

and so that a set of l image points $\mathbf{p}_i$ lies on $[u_{min}\ u_{max}] \times [v_{min}\ v_{max}]$ (where u_{min}, u_{max}, u_{min}, u_{max} are the image limits):

$$\begin{cases} u_i - u_{max} \leq 0\ ,\ i = 1 \cdots l, \\ u_{min} - u_i \leq 0\ ,\ i = 1 \cdots l, \\ v_i - v_{max} \leq 0\ ,\ i = 1 \cdots l, \\ v_{min} - v_i \leq 0\ ,\ i = 1 \cdots l. \end{cases} \tag{7.13}$$

The $4 \times l$ inequalities (7.13) can be converted to the following $2 \times l$ constraints:

$$\begin{cases} in_{2i-2}(\mathbf{x}) = u_i^2 - a_u u_i + b_u \leq 0 \ , \ i = 1 \cdots l, \\ in_{2i-1}(\mathbf{x}) = v_i^2 - a_v v_i + b_v \leq 0 \ , \ i = 1 \cdots l \end{cases} \tag{7.14}$$

where

$$\begin{cases} a_u = u_{max} + u_{min} \\ b_u = u_{max} u_{min}, \\ a_v = v_{max} + v_{min}, \\ b_v = v_{max} v_{min}. \end{cases}$$

The set of inequalities (7.14) can be converted to an equivalent set of equalities by introducing the $2l$ vector ζ of slack variables $\zeta = [\zeta_1 \cdots \zeta_{n_{in}}]^T$:

$$\begin{aligned} & in_i(\mathbf{x}, \mathbf{u}) \leq 0 \ \Leftrightarrow \\ & eq_i(\mathbf{x}, \zeta) = in_i(\mathbf{x}) + \zeta_i^2 = 0, \ i = 0 \cdots 2l - 1. \end{aligned} \tag{7.15}$$

Now, let us integrate the inequalities (7.15) with vectors of multipliers μ onto the following Hamiltonian:

$$H(\mathbf{X}, \dot{\mathbf{X}}) = \mathbf{U}^T \mathbf{U} + \lambda^T (\dot{\mathbf{x}} - \mathbf{U}) + + \mu^T \mathbf{eq}(\mathbf{x}, \zeta) \tag{7.16}$$

where $\mathbf{eq} = [eq_0 \cdots eq_{2l-1}]$ and $\mathbf{X}$ is an $18 + 4l$ extended state vector defined as

$$\mathbf{X} = \left[\mathbf{X}_x^T \ \mathbf{X}_u^T \ \mathbf{X}_\zeta^T \ \mathbf{X}_\lambda^T \ \mathbf{X}_\mu^T \right]^T \tag{7.17}$$

with

$$\begin{cases} \mathbf{X}_x = \mathbf{x}, \\ \dot{\mathbf{X}}_u = \mathbf{U}, \\ \dot{\mathbf{X}}_\zeta = [\zeta_1 \cdots \zeta_{2l}]^\top, \\ \dot{\mathbf{X}}_\lambda = [\lambda_1 \cdots \lambda_6]^\top, \\ \dot{\mathbf{X}}_\mu = [\mu_1 \cdots \mu_{2l}]^\top. \end{cases} \tag{7.18}$$

The solutions of the constrained optimal control problem defined by (7.11), (7.12), (7.13) can be related to those of an unconstrained variational calculus problem as proved in [13].

Proposition 7.2. *Vectors* $\mathbf{x}(t)$ *and* $\mathbf{U}(t)$ *satisfy the necessary conditions of the Minimum Principle for the optimal control problem defined by (7.11), (7.12), (7.13) if*

and only if there exists a piecewise smooth vector $\mathbf{X}(t)$ *defined in (7.17) which is a critical solution for the variational problem*

$$\min J(\mathbf{X}) = \int_{t_0}^{t_f} H(\mathbf{X}(t), \dot{\mathbf{X}}(t), t)dt \tag{7.19}$$

with the boundary conditions

$$\begin{cases} \mathbf{X}_x(t_0) = \mathbf{x}(t_0), \\ \mathbf{X}_u(t_0) = 0, \\ \mathbf{X}_\zeta(t_0) = 0, \\ \mathbf{X}_\lambda(t_0) = 0, \\ \mathbf{X}_\mu(t_0) = 0, \\ \mathbf{X}_x(t_f) = \mathbf{x}(t_f). \end{cases}$$

If $\mathbf{X}(t)$ is a solution for the variational problem (7.19) then, at all points where $\dot{\mathbf{X}}(t)$ is continuous, the Euler–Lagrange equation must hold:

$$\frac{d}{dt}\frac{\partial H}{\partial \dot{\mathbf{X}}} - \frac{\partial H}{\partial \mathbf{X}} = 0, \tag{7.20}$$

and at each discontinuity point of $\dot{\mathbf{X}}(t)$, the corner conditions must be satisfied:

$$\begin{aligned} \left(\frac{\partial H}{\partial \dot{\mathbf{X}}}\right)_{t^-} &= \left(\frac{\partial H}{\partial \dot{\mathbf{X}}}\right)_{t^+}, \\ \left(\dot{\mathbf{X}}\frac{\partial H}{\partial \dot{\mathbf{X}}} - H\right)_{t^-} &= \left(\dot{\mathbf{X}}\frac{\partial H}{\partial \dot{\mathbf{X}}} - H\right)_{t^+}. \end{aligned} \tag{7.21}$$

Admissible solution that satisfy the necessary conditions (7.20,7.21) are called critical solution. It can be shown that if $\mathbf{X}(t)$ is solution of the variational problem, then

$$\int_{t_0}^{t_f} (\mathbf{H}_X z + \mathbf{H}_{\dot{X}} \dot{z})\, dt = 0 \tag{7.22}$$

holds for all admissible piecewise smooth function $\mathbf{z}(t)$ that satisfies the boundary conditions ($\mathbf{H}_X$ and $\mathbf{H}_{\dot{X}}$ denote the partial derivatives of $\mathbf{H}$ with respect to $\mathbf{X}$ and $\dot{\mathbf{X}}$).

7.4 Solving the Variational Problem

Conventionally, the extremum of the variational problem are obtained by solving the Euler–Lagrange equation (7.20). However, these equations are only valid at the

extremum continuous points. At the points of discontinuities, the corners conditions (7.21) must be used. The positions, the number and the amplitudes of the points of discontinuities have then to be *a priori* known which make difficult the design of numerical method based on Euler–Lagrange equations. An alternative, based on the integral form (7.22) has been suggested by Gregory and Lin [13] and exploited by Zefran and Kumar in the context of two arms manipulation with inequality constraints [28]. The main idea is to choose particular admissible functions $z(t)$. First, the time interval $[t_0, t_f]$ is discretized in order that $t_0 < t_1 < \cdots < t_N = t_f$ and $t_k - t_{k-1} = h$ for $k = 1 \cdots N$. A set of piecewise functions are then defined:

$$\gamma_k(t) = \begin{cases} \dfrac{t - t_{k-1}}{h} \text{ if } t_{k-1} < t \leq t_k, \\ \dfrac{t_{k+1} - t}{h} \text{ if } t_k < t \leq t_{k+1}, \\ 0 \qquad \text{otherwise} \end{cases} \quad k = 0 \cdots N-1. \tag{7.23}$$

Since (7.22) is valid for all function and since $\mathbf{z}(t)$ is piecewise smooth, each of its components can be approximated with a sum $\widehat{\mathbf{z}}_i(t) = \sum_{j=0}^{N} z_{ij}\gamma_j(t)$. Using the central difference scheme to approximate the derivatives and mean-value theorem to approximate the integral (7.22) on each subinterval $[t_{k-1}, t_k]$, we obtain the following set of vectorial equations in the unknown values $\mathbf{X}_k$ at points $k = 1, \cdots, N-1$:

$$\frac{h}{2}\mathbf{H}_X(\mathbf{X}_{k-1}, \mathbf{X}_k) + \mathbf{H}_{\dot{X}}(\mathbf{X}_{k-1}, \mathbf{X}_k) + \frac{h}{2}\mathbf{H}_X(\mathbf{X}_k, \mathbf{X}_{k+1}) - \mathbf{H}_{\dot{X}}(\mathbf{X}_k, \mathbf{X}_{k+1}) = 0. \tag{7.24}$$

The resulting system of nonlinear equations is solved using the Newton–Raphson method. Each equation depends only on the three adjacent points. The matrix of the system of linear equations solved during the iteration is thus block-tridiagonal and the system can be solved very efficiently. After the numerical process, we obtain the extended state vector from which $\mathbf{x}(t)$ is extracted. The homography matrix and the collineation matrix can then be computed using (7.7) and (7.8).

In the next sections, we first show how trajectories in the image space can be obtained from the collineation matrix path. Our approach is then illustrated by examples of image synthesis.

7.5 Image Space Trajectories

In order to control efficiently a robot using visual data, we have to determine the trajectories of some image features in the image space. More precisely, we want to perform smooth trajectories $\mathbf{s}^*(t) = [x_1^*(t)\, y_1^*(t) \cdots x_n^*(t)\, y_n^*(t)]^T$ of n projected points in the image between a given start point $\mathbf{s}^*(0) = [x_1^*(0)\, y_1^*(0) \cdots x_n^*(0)\, y_n^*(0)]^T$ and a given desired point $\mathbf{s}^*(1) = [x_1^*(1)\, y_1^*(1) \cdots x_n^*(1)\, y_n^*(1)]^T$. We denote $\mathbf{p}_i^*(t) = [x_i^*(t)\, y_i^*(t)\, 1]^T$ the vector of homogeneous coordinates expressed in pixel of the

projection of a 3D point $\mathscr{X}_i$ in the current desired image (at time t). We define vector $\mathbf{h}_i = \alpha_i(t)\mathbf{p}_i^*(t)$. It is well-known that for all 3D points

$$\mathbf{h}_i(t) = \alpha_i(t)\mathbf{p}_i^*(t) = \mathbf{G}(t)\mathbf{p}_i^*(1) + \tau_i\gamma(t) \tag{7.25}$$

where $\alpha_i(t)$ is a positive scaling factor depending on time, τ_i is a constant scaling factor null if the target point belongs to Π, and $\gamma(t) = \mathbf{K}\mathbf{b}(t)$ represents the epipole in the current image (that is the projection in the image at time t of the optical center when the camera is in its desired position). After the initial collineation has been estimated, the optimal path of the collineation matrix can be computed as described previously. The initial value of the epipole, $\gamma(0) = \gamma_0$, can also be computed directly from image data (*i.e.*, γ_0 is independent of the $\mathbf{K}$-matrix) [10]. Furthermore, the epipole at time t can easily be computed from the state vector $\mathbf{x}(t)$. Note also that the scaling factor τ_i is not time dependent and can be computed directly from the initial and desired image data since (refer to (7.25))

$$\alpha_i(t)\mathbf{p}_i^*(0) \wedge \mathbf{p}_i^*(0) = 0 = \mathbf{G}(t)\mathbf{p}_i^*(1) \wedge \mathbf{p}_i^*(0) + \tau_i\gamma(t) \wedge \mathbf{p}_i^*(0).$$

We thus obtain[3]:

$$\tau_i = -\frac{(\mathbf{G}_0\mathbf{p}_i^*(1) \wedge \mathbf{p}_i^*(0))_1}{(\gamma_0 \wedge \mathbf{p}_i^*(0))_1}.$$

The trajectories of the considered point in the image corresponding to an optimal camera path can thus also be computed using

$$x_i^*(t) = \frac{(\mathbf{h}_i(t))_1}{(\mathbf{h}_i(t))_3},\ y_i^*(t) = \frac{(\mathbf{h}_i(t))_2}{(\mathbf{h}_i(t))_3}. \tag{7.26}$$

7.6 Example

The results have been obtained using an analog charge-coupled device (CCD) camera providing 640×480 images. Only the path-planning problem is addressed. Note however that the generated trajectories can be used in a visual servoing scheme as the one proposed in [22]. Our approach is illustrated by two experiments. The goal is to synthesize intermediate images between initial and desired images acquired by a CCD camera (boxed in Figures 7.4 and 7.5) corresponding to the solution of the variational problem previously presented. First, the optimal trajectory is computed without introducing the visibility constraints (7.13). As expected the camera follow a straight line (see Figure 7.2), however we observe that a part of the target get out of the camera field of view (see Figure 7.4 from time $t = 0.2$). In the second set of results, the visibility constraints is introduced in the variational problem. The optimization process is initialized using the unconstrained trajectories. The camera trajectory is no more a straight line (refer to Figure 7.3) and the target remains in the camera field of view (see Figure 7.5).

[3] $(\mathbf{v})_j$ denotes the j^{th} components of $\mathbf{v}$.

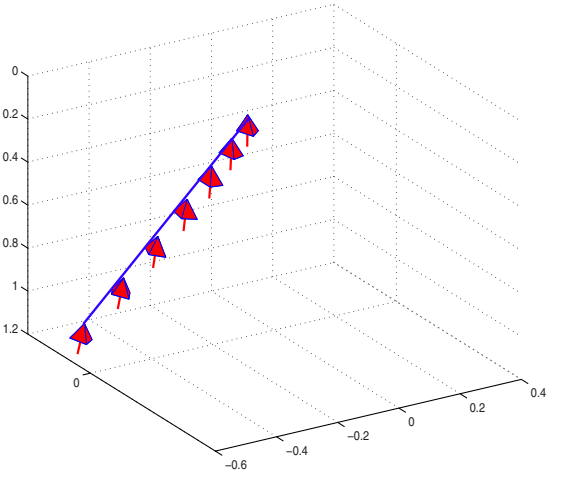


Fig. 7.2 Camera trajectories without visibility constraint

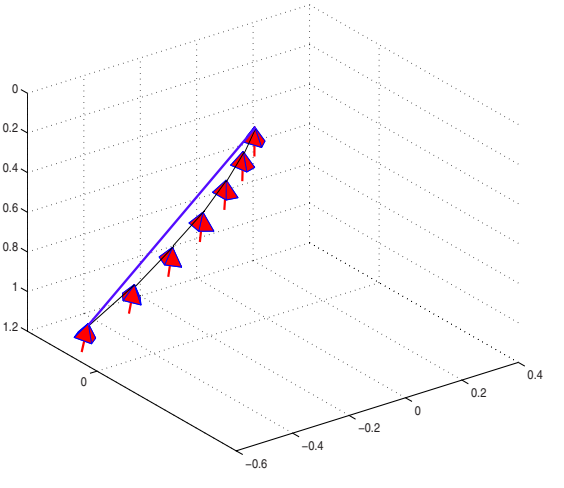


Fig. 7.3 Camera trajectories with visibility constraint

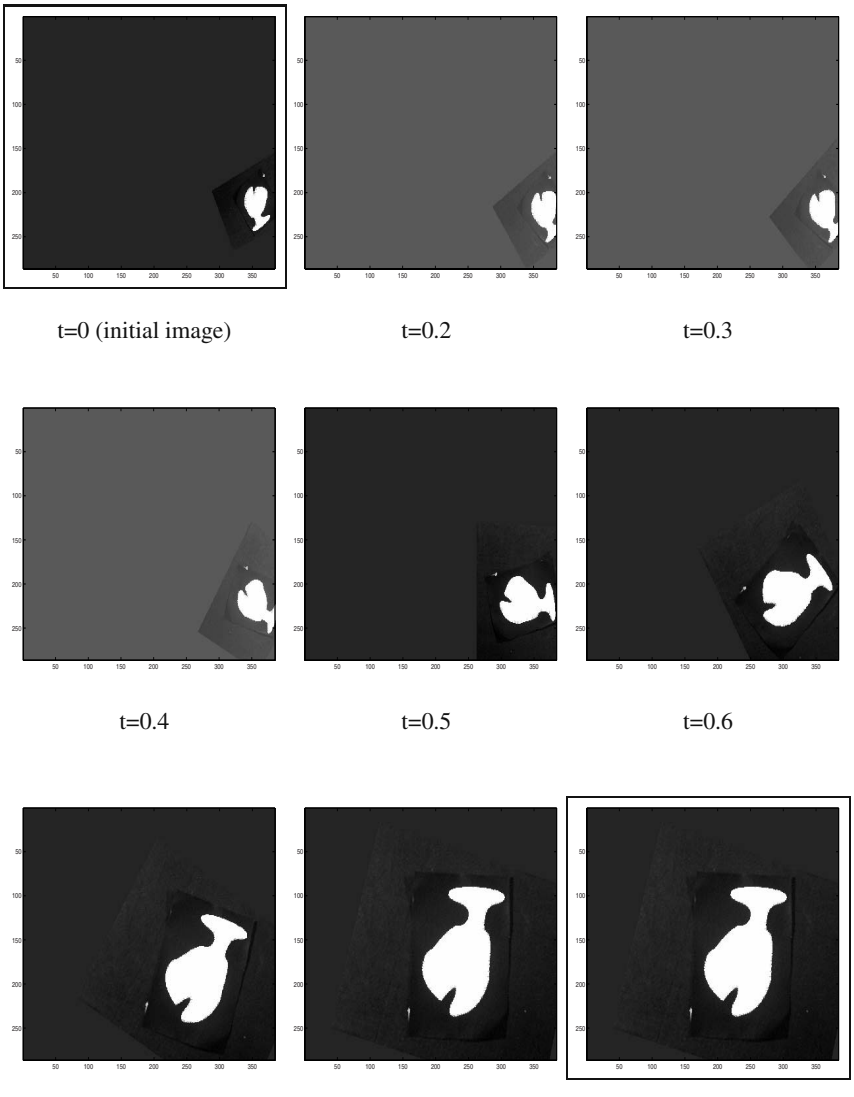


Fig. 7.4 Image trajectory without visibility constraint

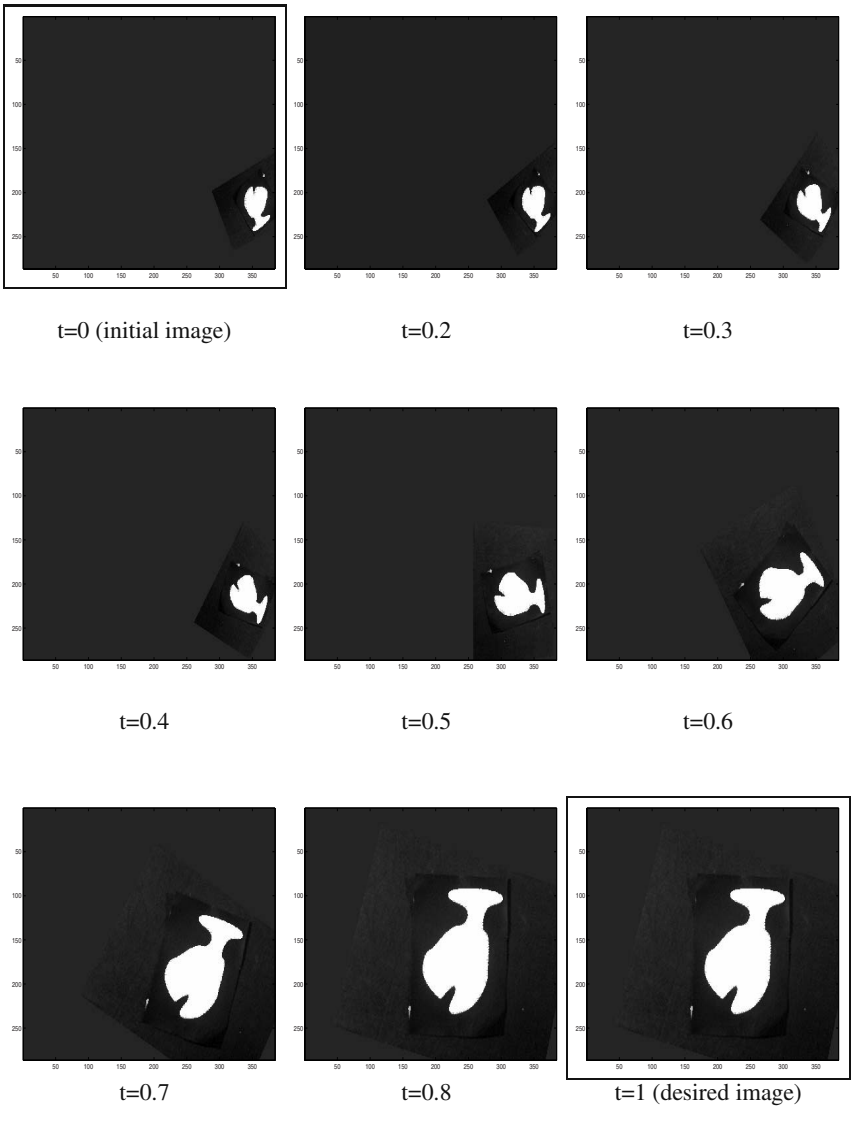


Fig. 7.5 Image trajectory under visibility constraint

7.7 Conclusion

In this chapter, we have addressed the problem of generating trajectories of some image features that corresponds to optimal 3D trajectories under visibility constraints. The obtained path can then be coupled to traditional image-based servoing in order to control efficiently a robotic system. It has first been shown that the path-planning problem can be formulated as an optimal control problem and then considered as a variational problem which can be solved using a numerical method.

References

[1] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman, D., Hager, G., Morse, A. (eds.) The Confluence of Vision and Control. LNCIS, vol. 237, pp. 66–78. Springer, Heidelberg (1998)
[2] Chaumette, F., Hutchinson, S.: Visual servo control, part i: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)
[3] Chaumette, F., Hutchinson, S.: Visual servo control, part ii: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)
[4] Chesi, G., Malis, E., Cipolla., R.: Automatic segmentation and matching of planar contours for visual servoing. In: International Conference on Robotics and Automation, San Francisco (2000)
[5] Chesi, G., Prattichizzo, D., Vicino, A.: Straight line path-planning in visual servoing. Journal of Dynamic Systems, Measurement and Control 129(4), 541–543 (2007)
[6] Chesi, G., Vicino, A.: Visual servoing for large camera displacements. IEEE Transactions on Robotics 20(4), 724–735 (2004)
[7] Corke, P.I., Hutchinson, S.A.: A new partitioned approach to image-based visual servo control. IEEE Transaction on Robotics and Automation 17(4), 507–515 (2001)
[8] Cowan, N., Lopes, G., Koditschek, D.: Rigid body visual servoing using navigation functions. In: IEEE Int. Conf. on Decision and Control, Sydney, Australia, pp. 3920–3926 (2000)
[9] Deguchi, K.: Optimal motion control for image-based visual servoing by decoupling translation and rotation. In: IEEE/RSJ International Conference on Intelligent Robots Systems, Victoria, BC, Canada, vol. 2, pp. 705–711 (1998)
[10] Faugeras, O.: Three-dimensionnal computer vision: a geometric view point. MIT Press, Cambridge (1993)
[11] Faugeras, O., Lustman, F.: Motion and structure from motion in a piecewise planar environment. Int. Journal of Pattern Recognition and Artificial Intelligence 2(3), 485–508 (1988)
[12] Fomena, R.T., Chaumette, F.: Visual servoing from spheres using a spherical projection model. In: IEEE International Conference on Robotics and Automation, ICRA 2007, Roma, Italia, pp. 2080–2085 (2007)
[13] Gregory, J., Lin, C.: Constrained optimization in the calculus of variations and optimal control theory. Van Nostrand Reinhold, New York (1992)
[14] Hamel, T., Mahony, R.: Visual servoing of an under-actuated dynamic rigid body system: an image-based approach. IEEE Transaction on Robotics and Automation 18(2), 187–198 (2002)
[15] Hashimoto, K., Noritugu, T.: Potential switching control in visual servo. In: IEEE Int. Conf. on Robotics and Automation, San Francisco, USA, pp. 2765–2770 (2000)
[16] Hosoda, K., Sakamoto, K., Asada, M.: Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems 1(3), 29–34 (1995)
[17] Iwatsuki, M., Okiyama, N.: A new formulation of visual servoing based on cylindrical coordinate system. IEEE Transactions on Robotics 21(2), 266–273 (2005)
[18] Lee, J.S., Suh, I., You, B.J., Oh, S.R.: A novel visual servoing approach involving disturbance observer. In: IEEE International Conference on Robotics and Automation, ICRA 1999, Detroit, Michigan, pp. 269–274 (1999)

[19] Mahony, R., Corke, P., Chaumette, F.: Choice of image features for depth-axis control in image-based visual servo control. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2002, Lausanne, Switzerland, vol. 1, pp. 390–395 (2002)
[20] Malis, E., Chaumette, F.: 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. International Journal of Computer Vision 37(1), 79–97 (2000)
[21] Malis, E., Chesi, G., Cipolla, R.: 2 1/2 d visual servoing with respect to planar contours having complex and unknown shapes. International Journal of Robotics Research, Special Issue on Visual Servoing 22(10-11), 841–853 (2003)
[22] Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Trans. on Robotics and Automation 18(4), 534–549 (2002)
[23] Mezouar, Y., Chaumette, F.: Optimal camera trajectory with image-based control. Int. Journal of Robotics Research, IJRR 22(10), 781–804 (2003)
[24] Mezouar, Y., Malis, E.: Robustness of central catadioptric image-based visual servoing to uncertainties on 3d parameters. In: IEEE/RSJ International Conference on Intelligent Robots Systems, pp. 1389–1394 (2004)
[25] Ruf, A., Horaud, R.: Visual trajectories from uncalibrated stereo. In: IEEE Int. Conf. on Intelligent Robots and Systems, pp. 83–91 (1997)
[26] Singh, R., Voyle, R.M., Littau, D., Papanikolopoulos, N.P.: Alignement of an eye-in-hand system to real objects using virtual images, Leuven, Belgium (1998)
[27] Tahri, O., Chaumette, F., Mezouar, Y.: New decoupled visual servoing scheme based on invariants from projection onto a sphere. In: IEEE International Conference on Robotics and Automation, ICRA 2008, Pasadena, California, USA, pp. 3238–3243 (2008)
[28] Zefran, M., Kumar., V.: Optimal control of systems with unilateral constraints. In: IEEE International Conference on Robotics and Automation, Nagoya, Japan, pp. 2695–2700 (1995)
[29] Zhang, H., Ostrowski, J.: Optimal motion planning in the image plane for mobile robots. In: IEEE Int. Conf. on Robotics and Automation, San Francisco, USA, pp. 3811–3816 (2000)
[30] Zhang, Z., Hanson, A.R.: Scaled euclidean 3d reconstruction based on externally uncalibrated cameras. In: IEEE Symposium on Computer Vision, Coral Gables, FL (1995)

Chapter 8
Estimation of Homography Dynamics on the Special Linear Group

Ezio Malis, Tarek Hamel, Robert Mahony, and Pascal Morin

Abstract. During the last decade, a number of highly successful visual servo control and real-time image processing algorithms have been proposed that use the homography between images of a planar scene as the primary measurement. The performance of the algorithms depends directly on the quality of the homography estimates obtained, and these estimates must be computed in real-time. In this chapter, we exploit the special linear Lie group structure of the set of all homographies to develop an on-line dynamic observer that provides smoothed estimates of a sequence of homographies and their relative velocities. The proposed observer is easy to implement and computationally undemanding. Furthermore, it is straightforward to tune the observer gains and excellent results are obtained for test sequences of simulation and real-world data.

8.1 Introduction

Visual servo control schemes use visual information obtained by one or multiple cameras as the primary measurement to regulate the motion of a robot [21, 11, 12, 6]. In the last decade, a number of visual servo control schemes have been proposed that extract homography image transformations between images of a planar scenes and use these as the primary visual information for a servo control problem [17, 7, 8].

Ezio Malis and Pascal Morin
INRIA, 2004 Route des Lucioles, 06902 Sophia Antipolis, France
e-mail: {ezio.malis,pascal.morin}@sophia.inria.fr

Tarek Hamel
I3S-CNRS, 2000 Route des Lucioles, 06903 Sophia Antipolis, France
e-mail: thamel@i3s.unice.fr

Robert Mahony
Department of Engineering, Australian National University, North Road,
Acton ACT 0200, Australia
e-mail: Robert.Mahony@anu.edu.au

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 133–150.
springerlink.com

A homography can be decomposed to explicitly reconstruct the pose (the translation and the rotation in Cartesian space) of the camera [9, 18] and the associated servo control task undertaken in Cartesian space [25, 16, 2, 24]. Alternatively, the control task can be defined in both image and Cartesian space; the rotation error is estimated explicitly and the translation error is expressed in the image [17, 7, 22, 8]. The resulting visual servo control algorithms are stable and robust [16] and do not depend on tracking of individual image features. Some recent work has been done on direct servo control of the homography matrix [1], an approach which offers considerable advantages in situations where the homography decomposition is ill-conditioned. A key component of this work is the identification of the group of homographies as a Lie group isomorphic to the special linear group $SL(3)$, an observation that has been known for some time in the computer vision community but had not been exploited before in the visual servo community.

In all cases, the performance of the closed-loop system depends on the quality of the homography estimates used as input to controller. In the case of visual servo control applications, the homographies must be computed in real-time with minimal computational overhead. Moreover, in such applications the homographies vary continuously and usually smoothly. It is natural, then, to consider using a dynamical observer (or filter) process in the closed-loop system to achieve temporal smoothing and averaging of the homography measurements. Such a process will reduce noise in the homography estimates, smoothing resulting closed-loop inputs and leading to improved performance, especially in visual servo applications. There has been a surge of interest recently in nonlinear observer design for systems with certain invariance properties [23, 5, 10, 15, 4] that have mostly been applied to applications in robotic vehicles [19, 20]. From these foundations there is an emerging framework for observer design for invariant systems on Lie groups [13, 3, 14]. The special linear group structure of the homographies [1] makes the homography observer problem an ideal application of these recent developments in observer theory.

In this chapter, we exploit the special linear Lie group structure of the set of all homographies to develop a dynamic observer to estimate homographies on-line. The proposed homography observer is based on constant velocity invariant kinematics on the Lie group. We assume that the velocity is unknown and propose an integral extension of the nonlinear observer to obtain estimates for both the homography and the velocity. We prove the existence of a Lyapunov function for the system, and use this to prove almost global stability and local exponential stability around the desired equilibrium point. The proposed algorithm provides high quality temporal smoothing of the homography data along with a smoothed homography velocity estimate. The estimation algorithm has been extensively tested in simulation and on real data.

The chapter is organized into five sections followed by a short conclusion. The present introduction section is followed by Section 8.2 that provides a recap of the Lie group structure of the set of homographies. The main contribution of the chapter is given in Section 8.3. Sections 8.4 and 8.5 provide an experimental study with simulated and real data.

8.2 Theoretical Background

A homography is a mapping between two images of a planar scene P. Let $p=(u,v)$ represent the pixel coordinates of a 3D point $\xi \in P$ as observed in the normalized image plane of a pinhole camera. Let $\mathscr{A}$ (resp. $\mathscr{B}$) denote projective coordinates for the image plane of a camera A (resp. B), and $\{A\}$ (resp. $\{B\}$) denote its frame of reference. A (3×3) homography matrix $H:\mathscr{A}\rightarrow\mathscr{B}$ defines the following mapping: $p^B = w(H,p^A)$, where

$$w(H,p)=\begin{bmatrix}(h_{11}u+h_{12}v+h_{13})/(h_{31}u+h_{32}v+h_{33})\\(h_{21}u+h_{22}v+h_{23})/(h_{31}u+h_{32}v+h_{33})\end{bmatrix}.$$

The mapping is defined up to a scale factor. That is, for any scaling factor $\mu\neq 0$, $p^B = w(\mu H,p^A) = w(H,p^A)$. The Lie group $SL(3)$ is the set of real matrices $SL(3)=\{H\in\mathbb{R}^{3\times3}\,\big|\,\det(H)=1\}$. If we suppose that the camera continuously observes the planar object, any homography can be represented by a homography matrix $H\in SL(3)$ such that

$$H=\gamma K\left(R+\frac{tn^\top}{d}\right)K^{-1} \tag{8.1}$$

where K is the upper triangular matrix containing the camera intrinsic parameters, R is the rotation matrix representing the orientation of $\{B\}$ with respect to $\{A\}$, t is the translation vector of coordinates of the origin of $\{B\}$ expressed in $\{A\}$, n is the normal to the planar surface P expressed in $\{A\}$, d is the orthogonal distance of the origin of $\{A\}$ to the planar surface, and γ is a scaling factor:

$$\gamma=\det\left(R+\frac{tn^\top}{d}\right)^{-\frac{1}{3}}=\left(1+\frac{n^\top R^\top t}{d}\right)^{-\frac{1}{3}}.$$

Correspondingly, knowing the camera intrinsic parameters matrix K, any full rank 3×3 matrix with unitary determinant can be decomposed according to (8.1) (see [9] for a numerical decomposition and [18] for the analytical decomposition). Note that there exist two possible solutions to the decomposition. The planar surface P is parametrized by

$$P=\{\xi\in\{A\}\mid n^\top\xi=d\}$$

For any two frames $\{A\}$ and $\{B\}$ whose origins lie on the same side of the planar surface P then $n^\top Rt>-d$ by construction and the determinant of the associated homography $\det(H)=1$.

The map w is a group action of $SL(3)$ on R^2:

$$w(H_1,w(H_2,p))=w(H_1H_2,p)$$

where H_1,H_2 and $H_1H_2\in SL(3)$. The geometrical meaning of this property is that the 3D motion of the camera between views $\{A\}$ and $\{B\}$, followed by the 3D

motion between views $\{B\}$ and $\{C\}$ is the same as the 3D motion between views $\{A\}$ and $\{C\}$.

Remark 8.1. The local parametrization given by (8.1) is singular when $\{A\}$ and $\{B\}$ are collocated. That is, when $t = 0$, the differential of the mapping defined by (8.1) is degenerate. Indeed, in this case the normal to the plane n is not observable. The singularity of the parametrization does not affect the validity of the correspondence $\mathscr{H} \equiv SL(3)$, however, it does mean that the parametrization (8.1) is very poorly conditioned for homography matrices close to $SO(3)$. This is fundamental reasons why it is preferable to do both image based visual servo control and temporal smoothing directly on the homography group rather than extracting structure variables explicitly.

The Lie algebra $\mathfrak{sl}(3)$ for $SL(3)$ is the set of matrices with trace equal to zero: $\mathfrak{sl}(3) = \{X \in \mathbb{R}^{3\times 3} \mid \mathrm{tr}(X) = 0\}$. The adjoint operator is a mapping $\mathrm{Ad} : SL(3) \times \mathfrak{sl}(3) \rightarrow \mathfrak{sl}(3)$ defined by

$$\mathrm{Ad}_H X = HXH^{-1}, \quad H \in SL(3), X \in \mathfrak{sl}(3).$$

For any two matrices $A, B \in \mathbb{R}^{3\times 3}$ the Euclidean matrix inner product and Frobenius norm are defined as

$$\langle\langle A, B\rangle\rangle = \mathrm{tr}(A^T B), \qquad ||A|| = \sqrt{\langle\langle A, A\rangle\rangle}.$$

Let $\mathbb{P}$ denote the unique orthogonal projection of $\mathbb{R}^{3\times 3}$ onto $\mathfrak{sl}(3)$ with respect to the inner product $\langle\langle \cdot, \cdot \rangle\rangle$,

$$\mathbb{P}(H) := \left(H - \frac{\mathrm{tr}(H)}{3} I\right) \in \mathfrak{sl}(3). \tag{8.2}$$

The projection onto the complementary subspace (the span of I in $\mathbb{R}^{3\times 3}$) is defined by

$$\mathbb{P}^{\perp}(H) := H - \mathbb{P}(H) = \frac{\mathrm{tr}(H)}{3} I. \tag{8.3}$$

Clearly one has $\langle\langle \mathbb{P}(H), \mathbb{P}^{\perp}(H)\rangle\rangle = 0$.

8.3 Nonlinear Observer on $SL(3)$

Consider the left invariant kinematics defined on $SL(3)$

$$\dot{H} = HA \tag{8.4}$$

where $H \in SL(3)$ and $A \in \mathfrak{sl}(3)$. A general framework for nonlinear filtering on the special linear group is introduced. The theory is developed for the case where A is assumed to be unknown and constant. The goal is to provide a set of dynamics for an estimate $\widehat{H}(t) \in SL(3)$ of $H(t)$ and an estimate $\widehat{A}(t) \in \mathfrak{sl}(3)$ of A to drive

the estimation error $\widetilde{H} = \widehat{H}^{-1}H$ to the identity matrix I, and the estimation error $\widetilde{A} = A - \widehat{A}$ to zero.

The estimator filter equation of $\widehat{H}$ is posed directly on $SL(3)$. It includes a correction term derived from the error $\widetilde{H}$. We consider an estimator filter of the form

$$\begin{cases} \dot{\widehat{H}} = \widehat{H}\left(\mathrm{Ad}_{\widehat{H}}\widehat{A} + \alpha(\widehat{H},H)\right), \quad \widehat{H}(0) = \widehat{H}_0, \\ \dot{\widehat{A}} = \beta(\widehat{H},H), \quad \widehat{A}(0) = \widehat{A}_0. \end{cases} \tag{8.5}$$

This yields the following expression for the dynamics of the estimation error $(\widetilde{H},\widetilde{A}) = (\widehat{H}^{-1}H, A - \widehat{A})$:

$$\begin{cases} \dot{\widetilde{H}} = \widetilde{H}\left(\widetilde{A} - \mathrm{Ad}_{\widetilde{H}^{-1}}\alpha\right) \\ \dot{\widetilde{A}} = -\beta \end{cases} \tag{8.6}$$

with the arguments of α and β omitted to lighten the notation. The main result of the chapter is stated next.

Theorem 8.1. *Assume that the matrix A in* (8.4) *is constant. Consider the nonlinear estimator filter* (8.5) *along with the innovation α and the estimation dynamics β defined as*

$$\begin{cases} \alpha = -k_H \mathrm{Ad}_{\widetilde{H}}\mathbb{P}(\widetilde{H}^T(I - \widetilde{H})), \quad k_H > 0 \\ \beta = -k_A \mathbb{P}(\widetilde{H}^T(I - \widetilde{H})), \quad k_A > 0 \end{cases} \tag{8.7}$$

with the projection operator $\mathbb{P} : \mathbb{R}^{3\times 3} \to \mathfrak{sl}(3)$ defined by (8.2)*. Then, for the estimation error dynamics* (8.6)*,*

i) *all solutions converge to $E = E_s \cup E_u$ with:*

$$\begin{aligned} E_s &= (I,0) \\ E_u &= \{(\widetilde{H}_0,0) | \widetilde{H}_0 = \lambda(I + (\lambda^{-3} - 1)vv^\top), v \in \mathbb{S}^2\} \end{aligned}$$

where $\lambda \approx -0.7549$ is the unique real solution of the equation $\lambda^3 - \lambda^2 + 1 = 0$;

ii) *the equilibrium point $E_s = (I,0)$ is locally exponentially stable;*

iii) *any point of E_u is an unstable equilibrium. More precisely, for any $(\widetilde{H}_0,0) \in E_u$ and any neighborhood $\mathscr{U}$ of $(\widetilde{H}_0,0)$, there exists $(\widetilde{H}_1,\widetilde{A}_1) \in \mathscr{U}$ such that the solution of system* (8.6) *issued from $(\widetilde{H}_1,\widetilde{A}_1)$ converges to E_s.*

Proof. Let us prove part *i)*. Let us consider the following candidate Lyapunov function

$$\begin{aligned} V(\widetilde{H},\widetilde{A}) &= \frac{1}{2}\|I - \widetilde{H}\|^2 + \frac{1}{2k_A}\|\widetilde{A}\|^2 \\ &= \frac{1}{2}\mathrm{tr}((I - \widetilde{H})^T(I - \widetilde{H})) + \frac{1}{2k_A}\mathrm{tr}(\widetilde{A}^T\widetilde{A}). \end{aligned} \tag{8.8}$$

The derivative of V along the solutions of system (8.6) is

$$\begin{aligned}\dot{V} &= -\mathrm{tr}((I-\widetilde{H})^T\dot{\widetilde{H}}) + \frac{1}{k_A}\mathrm{tr}(\widetilde{A}^T\dot{\widetilde{A}}) \\ &= -\mathrm{tr}((I-\widetilde{H})^T\widetilde{H}\widetilde{A} - (I-\widetilde{H})^T\widetilde{H}\mathrm{Ad}_{\widetilde{H}^{-1}}\alpha) - \frac{1}{k_A}\mathrm{tr}(\widetilde{A}^T\beta).\end{aligned}$$

Knowing that for any matrices $G \in SL(3)$ and $B \in \mathfrak{sl}(3)$, $\mathrm{tr}(B^TG) = \mathrm{tr}(B^T\mathbb{P}(G)) = \langle\langle B, \mathbb{P}(G)\rangle\rangle$, one obtains:

$$\dot{V} = \langle\langle \mathbb{P}(\widetilde{H}^T(I-\widetilde{H})), \mathrm{Ad}_{\widetilde{H}^{-1}}\alpha\rangle\rangle - \langle\langle \widetilde{A}, \mathbb{P}(\widetilde{H}^T(I-\widetilde{H})) + \frac{1}{k_A}\beta\rangle\rangle.$$

Introducing the expressions of α and β (8.7) in the above equation yields

$$\dot{V} = -k_H||\mathbb{P}(\widetilde{H}^T(I-\widetilde{H}))||^2. \tag{8.9}$$

The derivative of the Lyapunov function is negative semidefinite, and equal to zero when $\mathbb{P}(\widetilde{H}^T(I-\widetilde{H})) = 0$. The dynamics of the estimation error is autonomous, *i.e.* it is given by

$$\begin{cases}\dot{\widetilde{H}} = \widetilde{H}\left(\widetilde{A} + k_H\mathbb{P}(\widetilde{H}^T(I-\widetilde{H}))\right) \\ \dot{\widetilde{A}} = k_A\mathbb{P}(\widetilde{H}^T(I-\widetilde{H})).\end{cases} \tag{8.10}$$

Therefore, we deduce from LaSalle theorem that all solutions of this system converge to the largest invariant set contained in $\{(\widetilde{H},\widetilde{A})|\mathbb{P}(\widetilde{H}^T(I-\widetilde{H})) = 0\}$.

We now prove that, for system (8.10), the largest invariant set E contained in $\{(\widetilde{H},\widetilde{A})|\mathbb{P}(\widetilde{H}^T(I-\widetilde{H})) = 0\}$ is equal to $E_s \cup E_u$. We need to show that the solutions of system (8.10) belonging to $\{(\widetilde{H},\widetilde{A})|\mathbb{P}(\widetilde{H}^T(I-\widetilde{H})) = 0\}$ for all t consist of all fixed points of $E_s \cup E_u$. Note that $E_s = (I,0)$ is clearly contained in E. Let us thus consider such a solution $(\widetilde{H}(t),\widetilde{A}(t))$. First, we deduce from (8.10) that $\dot{\widetilde{A}}(t)$ is identically zero since $\mathbb{P}(\widetilde{H}^T(t)(I-\widetilde{H}(t)))$ is identically zero on the invariant set E and therefore $\widetilde{A}$ is constant. We also deduce from (8.10) that $\widetilde{H}$ is solution to the equation $\dot{\widetilde{H}} = \widetilde{H}\widetilde{A}$. Note that at this point one cannot infer that $\widetilde{H}$ is constant. Still, we omit from now on the possible time-dependence of $\widetilde{H}$ to lighten the notation.

Since $\mathbb{P}(\widetilde{H}^T(I-\widetilde{H})) = 0$, we have that

$$\widetilde{H}^\top(I-\widetilde{H}) = \frac{1}{3}\mathrm{trace}(\widetilde{H}^\top(I-\widetilde{H}))I \tag{8.11}$$

which means that $\widetilde{H}$ is a symmetric matrix. Therefore, it can be decomposed as:

$$\widetilde{H} = UDU^\top \tag{8.12}$$

where $U \in SO(3)$ and $D = \mathrm{diag}(\lambda_1,\lambda_2,\lambda_3) \in SL(3)$ is a diagonal matrix which contains the three real eigenvalues of $\widetilde{H}$. Without loss of generality let us suppose that the eigenvalues are in increasing order: $\lambda_1 \leq \lambda_2 \leq \lambda_3$.

Plugging (8.12) into (8.11), one obtains:

$$D(I-D) = \frac{1}{3}\text{trace}(D(I-D))I.$$

Knowing that $\det(D) = 1$, the λ_i's satisfy the following equations:

$$\lambda_1(1-\lambda_1) = \lambda_2(1-\lambda_2) \tag{8.13}$$
$$\lambda_2(1-\lambda_2) = \lambda_3(1-\lambda_3) \tag{8.14}$$
$$\lambda_3 = 1/(\lambda_1\lambda_2) \tag{8.15}$$

which can also be written as follows:

$$\lambda_1 - \lambda_2 = (\lambda_1 - \lambda_2)(\lambda_1 + \lambda_2) \tag{8.16}$$
$$\lambda_1 - \lambda_3 = (\lambda_1 - \lambda_3)(\lambda_1 + \lambda_3) \tag{8.17}$$
$$\lambda_3 = 1/(\lambda_1\lambda_2). \tag{8.18}$$

First of all, let us remark that if $\lambda_1 = \lambda_2 = \lambda_3$ then $\lambda_1 = \lambda_2 = \lambda_3 = 1$. This solution is associated with the equilibrium point $E_s = (I, 0)$.

If $\lambda_1 = \lambda_2 < \lambda_3$ then:

$$1 = \lambda_2 + \lambda_3 \tag{8.19}$$
$$\lambda_3 = 1/(\lambda_2^2) \tag{8.20}$$

where $\lambda_2 \approx -0.7549$ is the unique real solution of the equation $\lambda_2^3 - \lambda_2^2 + 1 = 0$. This solution is associated with the equilibrium set E_u.

If $\lambda_1 < \lambda_2 = \lambda_3$ then:

$$1 = \lambda_1 + \lambda_2 \tag{8.21}$$
$$\lambda_1 = 1/\lambda_2^2 \tag{8.22}$$

so that λ_2 is also solution of the equation $\lambda_2^3 - \lambda_2^2 + 1 = 0$. But this is impossible since we supposed $\lambda_1 < \lambda_2$ and the solution of the equation is such that $-1 < \lambda_2 < 0$ and $0 < \lambda_1 = 1/\lambda_2^2 < 1$.

If $\lambda_1 \neq \lambda_2 \neq \lambda_3$, then:

$$1 = \lambda_1 + \lambda_2 \tag{8.23}$$
$$1 = \lambda_1 + \lambda_3 \tag{8.24}$$
$$\lambda_3 = 1/(\lambda_1\lambda_2) \tag{8.25}$$

which means that $\lambda_2 = \lambda_3$. This is in contradiction with our initial hypothesis.

In conclusion, $\widetilde{H}$ has two equal negative eigenvalues $\lambda_1 = \lambda_2 = \lambda$ ($\lambda \approx -0.7549$ is the unique real solution of the equation $\lambda^3 - \lambda^2 + 1 = 0$) and the third one is $\lambda_3 = 1/\lambda^2$. Writing the diagonal matrix D as follows:

$$D = \lambda(I + (\lambda^{-3} - 1)e_3 e_3^\top)$$

and plugging this equation into (8.12), the homography for the second solution ($\lambda_1 = \lambda_2$) can be expressed as follows:

$$\widetilde{H} = \lambda(I + (\lambda^{-3} - 1)(Ue_3)(Ue_3)^\top).$$

Setting $v = Ue_3$, we finally find that $\widetilde{H}$ must have the following form:

$$\widetilde{H} = \lambda(I + (\lambda^{-3} - 1)vv^\top)$$

where v is a unitary vector: $\|v\| = 1$ and λ is the unique real constant value that verifies the equation $\lambda^3 - \lambda^2 + 1 = 0$.

It remains to show that $\widetilde{A} = 0$. The inverse of $\widetilde{H}$ is

$$\widetilde{H}^{-1} = \lambda^{-1}(I + (\lambda^3 - 1)vv^\top).$$

The derivative of $\widetilde{H}$ is

$$\dot{\widetilde{H}} = \lambda(\lambda^{-3} - 1)(\dot{v}v^\top + v\dot{v}^\top)$$

so that

$$\widetilde{A} = \widetilde{H}^{-1}\dot{\widetilde{H}} = (\lambda^{-3} - 1)(I + (\lambda^3 - 1)vv^\top)(\dot{v}v^\top + v\dot{v}^\top).$$

Knowing that $v^\top\dot{v} = 0$, this equation becomes:

$$\widetilde{A} = (\lambda^{-3} - 1)(\dot{v}v^\top + v\dot{v}^\top + (\lambda^3 - 1)v\dot{v}^\top)$$

and knowing that $\lambda^3 = \lambda^2 - 1$, we obtain:

$$\widetilde{A} = (\lambda^{-3} - 1)(\dot{v}v^\top + v\dot{v}^\top + (\lambda^2 - 2)v\dot{v}^\top) \tag{8.26}$$

$$= (\lambda^{-3} - 1)(\dot{v}v^\top - v\dot{v}^\top + \lambda^2 v\dot{v}^\top). \tag{8.27}$$

Since $\dot{v}v^\top - v\dot{v}^\top = [v \times \dot{v}]_\times = [[v]_\times \dot{v}]_\times$[1], we finally obtain

$$\widetilde{A} = (\lambda^{-3} - 1)([[v]_\times \dot{v}]_\times + \lambda^2 v\dot{v}^\top).$$

Since $[[v]_\times \dot{v}]_\times$ is a skew-symmetric matrix, the diagonal elements of $\widetilde{A}$ are $a_{ii} = (\lambda^{-3} - 1)\lambda^2 v_i \dot{v}_i$. Knowing that each a_{ii} is constant we have two possible cases. The first one is $a_{ii} = 0$ for each i. Then v is constant so that $\widetilde{H}$ is also constant and $\widetilde{A} = 0$. If there exists i such that $a_{ii} \neq 0$, then there exists i such that $a_{ii} < 0$. This is due to the fact that $\widetilde{A} \in \mathfrak{sl}(3)$ and therefore $\sum_i a_{ii} = 0$. In this case, the corresponding v_i diverges to infinity because $v_i \dot{v}_i$ is a strictly positive constant. This contradicts the fact that $\|v\| = 1$. This concludes the proof of part *i)* of the theorem.

Let us prove part *ii)*. We compute the linearization of system (8.10) at $E_s = (I, 0)$. Let us define X_1 and X_2 as elements of $\mathfrak{sl}(3)$ corresponding to the first order approximations of $\widetilde{H}$ and $\widetilde{A}$ around $(I, 0)$:

[1] $[v]_\times$ represents the skew-symmetric matrix associated with the cross-product by $v \in \mathbb{R}^3$, *i.e.* $[v]_\times y = v \times y$, $\forall y \in \mathbb{R}^3$.

$$\widetilde{H} \approx (I + X_1), \qquad \widetilde{A} \approx X_2.$$

Substituting these approximations into (8.10) and discarding all terms quadratic or higher order in (X_1, X_2) yields

$$\begin{pmatrix} \dot{X}_1 \\ \dot{X}_2 \end{pmatrix} = \begin{pmatrix} -k_H I_3 & I_3 \\ -k_A I_3 & 0 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}. \tag{8.28}$$

Since $k_H, k_A > 0$, the linearized error system is exponentially stable. This proves the local exponential stability of the equilibrium $(I, 0)$.

Lastly, let us prove part *iii)*. First, we remark that the function V is constant and strictly positive on the set E_u. This can be easily verified from (8.8) and the definition of E_u, using the fact that on this set $\widetilde{A} = 0$, $\widetilde{H}^T \widetilde{H} = \widetilde{H}^2 = \lambda^2 I + (\frac{1}{\lambda^2} - \lambda) v v^T$, and $\mathrm{tr}(vv^T) = 1$ since $\|v\| = 1$. We denote by V_u the value of V on E_u. The fact that V_u is strictly positive readily implies (in accordance with part *ii)*) that E_s is an asymptotically stable equilibrium, since V is nonincreasing along the system's solutions, and each of them converges to $E_s \cup E_u$. Using the same arguments, the proof of part *iii)* reduces to showing that for any point $(\widetilde{H}_0, 0) \in E_u$, and any neighborhood $\mathscr{U}$ of this point, one can find $(\widetilde{H}_1, \widetilde{A}_1) \in \mathscr{U}$ such that

$$V(\widetilde{H}_1, \widetilde{A}_1) < V_u. \tag{8.29}$$

Let $\widetilde{H}(.)$ denote a smooth curve on $SL(3)$, solution of $\dot{\widetilde{H}} = \widetilde{H}C$ with C a constant element of $\mathfrak{sl}(3)$ that will be specified latter on. We also assume that $(\widetilde{H}(0), 0) \in E_u$. Let $f(t) = \|I - \widetilde{H}(t)\|^2/2$ so that, by (8.8), $f(0) = V_u$. The first derivative of f is given by

$$\begin{aligned} \dot{f}(t) &= -\mathrm{tr}((I - \widetilde{H}(t))^T \dot{\widetilde{H}}(t)) \\ &= -\mathrm{tr}((I - \widetilde{H}(t))^T \widetilde{H}(t) C) \\ &= -\langle\langle \mathbb{P}(\widetilde{H}^T(t)(I - \widetilde{H}(t))), C \rangle\rangle. \end{aligned}$$

For all elements $(\widetilde{H}_0, 0) \in E_u$, one has $\mathbb{P}(\widetilde{H}_0^T (I - \widetilde{H}_0)) = 0$, so that $\dot{f}(0) = 0$. We now calculate the second order derivative of f:

$$\begin{aligned} \ddot{f}(t) &= \mathrm{tr}(\dot{\widetilde{H}}(t)^T \dot{\widetilde{H}}(t)) - \mathrm{tr}((I - \widetilde{H}(t))^T \ddot{\widetilde{H}}(t)) \\ &= \mathrm{tr}(\dot{\widetilde{H}}(t)^T \dot{\widetilde{H}}(t)) - \mathrm{tr}\left((I - \widetilde{H}(t))^T \dot{\widetilde{H}}(t) C\right) \end{aligned}$$

where we have used the fact that C is constant. Evaluating the above expression at $t = 0$ and replacing $\dot{\widetilde{H}}(0)$ by its value $\widetilde{H}(0)C$ yields

$$\ddot{f}(0) = \|\widetilde{H}(0)C\|^2 - \mathrm{tr}\left((I - \widetilde{H}(0))^T \widetilde{H}(0) C^2\right). \tag{8.30}$$

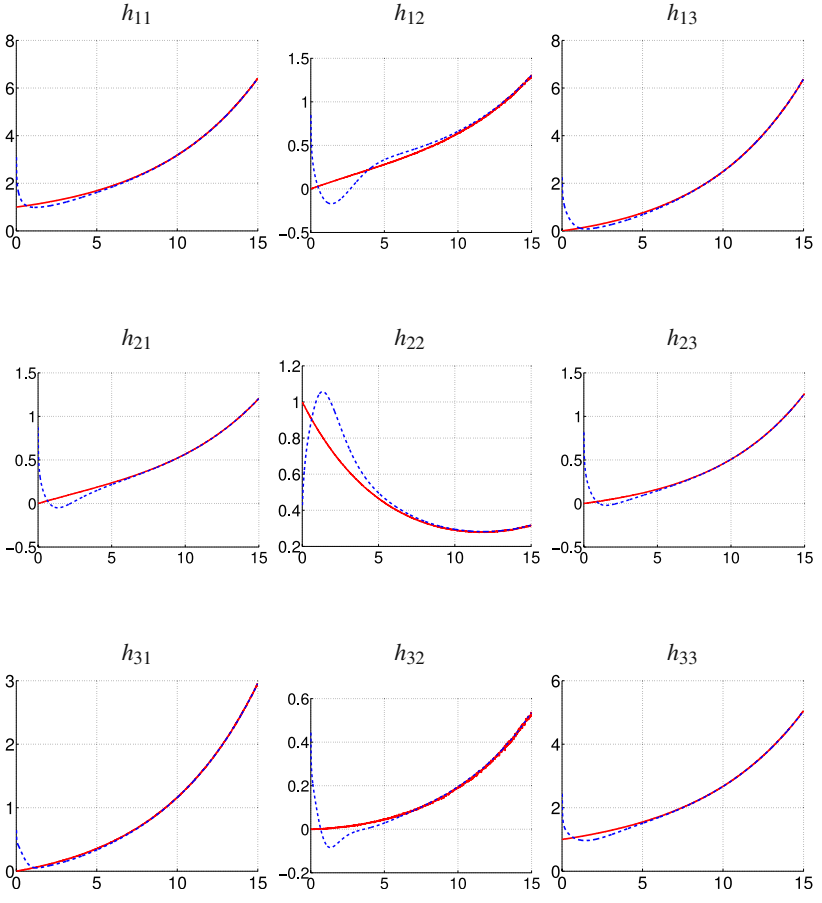


Fig. 8.1 Plain line: the measured homography matrix H. Dashed line: the observed homography $\widehat{H}$

When $(\widetilde{H}_0, 0) \in E_u$, one has

$$\widetilde{H}_0^2 = \lambda^2 I + (\frac{1}{\lambda^2} - \lambda) vv^T = \widetilde{H}_0 + (\lambda^2 - \lambda) I.$$

Therefore, we deduce from (8.30) that

$$\ddot{f}(0) = \|\widetilde{H}(0)C\|^2 + \lambda(\lambda - 1)\mathrm{tr}(C^2). \tag{8.31}$$

Since $(\widetilde{H}(0), 0) \in E_u$, there exists $v \in \mathbb{S}^2$ such that $\widetilde{H}(0) = \lambda I + (\frac{1}{\lambda^2} - \lambda) vv^T$. From this expression and using the fact that $\lambda^3 - \lambda^2 + 1 = 0$, one verifies that

$$\|\widetilde{H}(0)C\|^2 = \lambda^2 \|C\|^2 + (\frac{1}{\lambda^2} - \lambda)\mathrm{tr}(C^T vv^T C). \tag{8.32}$$

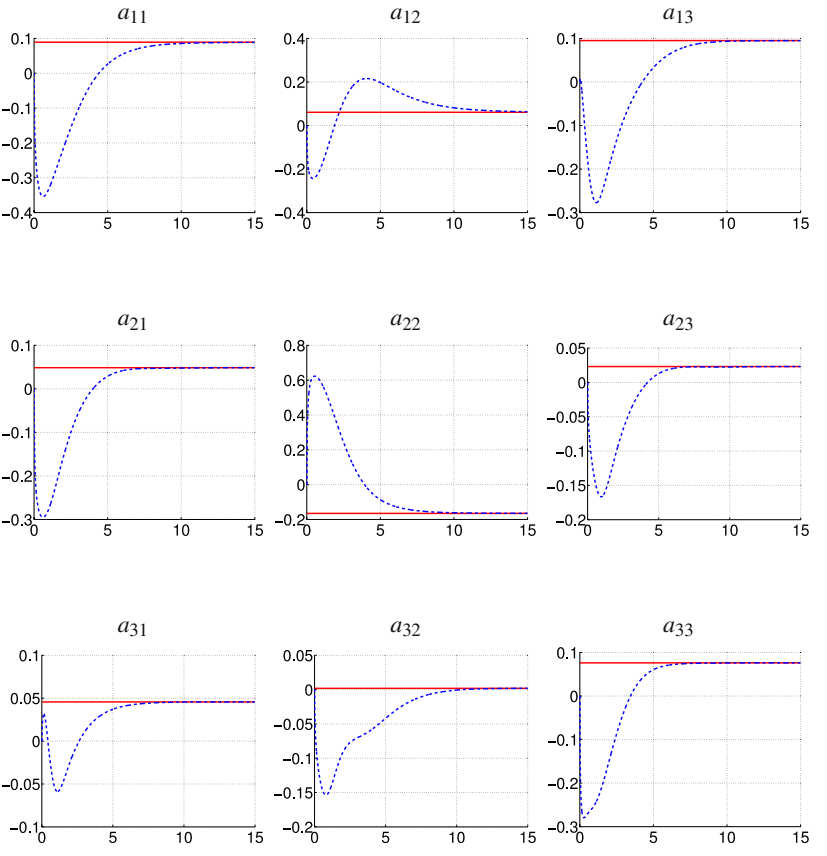


Fig. 8.2 Plain line: the true homography velocity A. Dashed line the observed homography velocity $\widehat{A}$

Now let us set $C = [v]_\times$, the skew-symmetric matrix associated with the vector v. Clearly, $C \in \mathfrak{sl}(3)$. Then, it follows from (8.31) and (8.32) that

$$\begin{aligned}\ddot{f}(0) &= \lambda^2\|C\|^2 + \lambda(\lambda-1)\mathrm{tr}(C^2)\\ &= \lambda^2\mathrm{tr}(v_\times^T v_\times) + \lambda(\lambda-1)\mathrm{tr}((v_\times)^2)\\ &= -\lambda^2\mathrm{tr}((v_\times)^2) + \lambda(\lambda-1)\mathrm{tr}((v_\times)^2)\\ &= -\lambda\,\mathrm{tr}((v_\times)^2) = 2\lambda\|v\|^2 = 2\lambda < 0.\end{aligned}$$

Therefore, there exists $t_1 > 0$ such that for any $t \in (0, t_1)$,

$$\begin{aligned}f(t) &\approx f(0) + t\dot{f}(0) + t^2/2\ddot{f}(0)\\ &\approx V_u + t^2/2\ddot{f}(0) < V_u.\end{aligned}$$

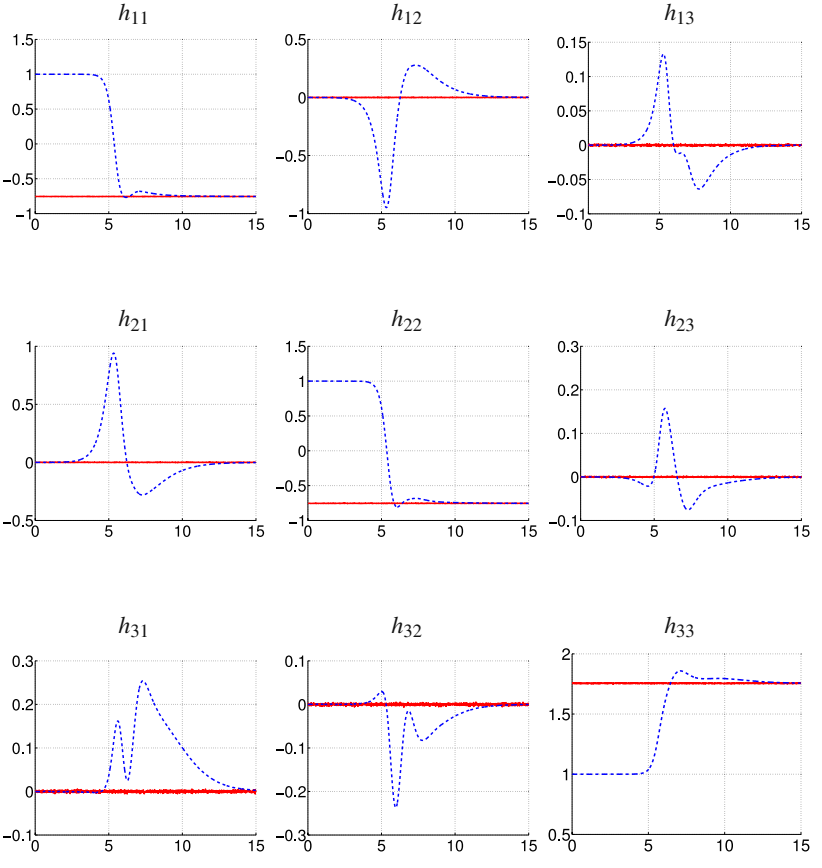


Fig. 8.3 Instability of critical points. Plain line: the measured homography matrix H. Dashed line: the observed homography $\widehat{H}$

Equation 8.29 follows by setting $(\widetilde{H}_1, A_1) = (\widetilde{H}(t), 0)$ with $t \in (0, t_1)$ chosen small enough so as to have $(\widetilde{H}(t), 0) \in \mathscr{U}$. This concludes the proof of part *iii)* and the proof of the theorem. □

8.4 Simulations with Ground Truth

We validated the proposed observer with several simulations. In this section, we illustrate and discuss two simulations results. We use the known ground truth to assess the quality of the homography and velocity estimations.

In order to simulate a real experiment, we build a sequence of reference homographies, starting from an initial homography $H_0 \in SL(3)$. The reference set of homographies was built using the following formula:

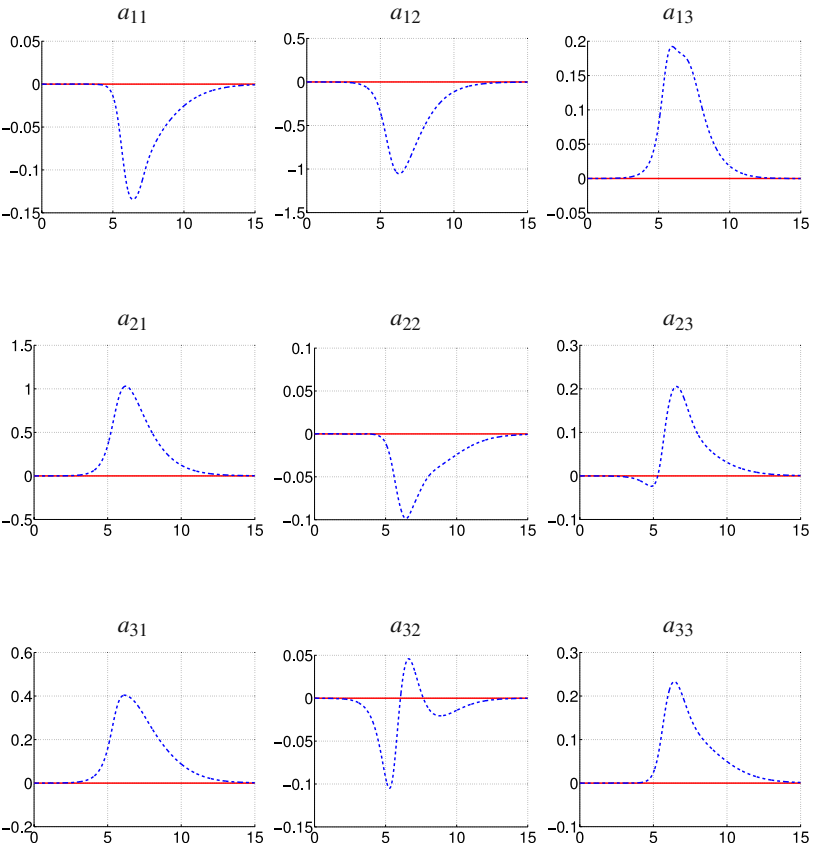


Fig. 8.4 Instability of critical points. Plain line: the true homography velocity A. Dashed line the observed homography velocity $\widehat{A}$

$$H_{k+1} = H_k \exp(A\Delta t + Q_k \Delta t)$$

where $A \in \mathfrak{sl}(3)$ is a constant velocity, $Q_k \in \mathfrak{sl}(3)$ is a random matrix with Gaussian distribution, and Δt is the sampling time (in the simulation we set the variance to $\sigma = 0.1$). By building the homographies in this way, we guarantee that the measured $H_k \in SL(3), \forall k$.

We implemented a discretized observer in order to process the data. In all examples the gains of the observer were set to $k_H = 2$ and $k_A = 1$.

Figure 8.1 shows the elements of the measured and estimated homography matrices. Figure 8.2 shows the elements of the associated homography velocities. In this simulation the initial "error" for the homography is chosen at random and it is very large. The initial velocity estimate $\widehat{A}_0$ is set to zero. Figure 8.1 shows that after a fast transient the estimated homography converges towards the measured homography. Figure 8.2 shows that the estimated velocity also converges towards the true one.

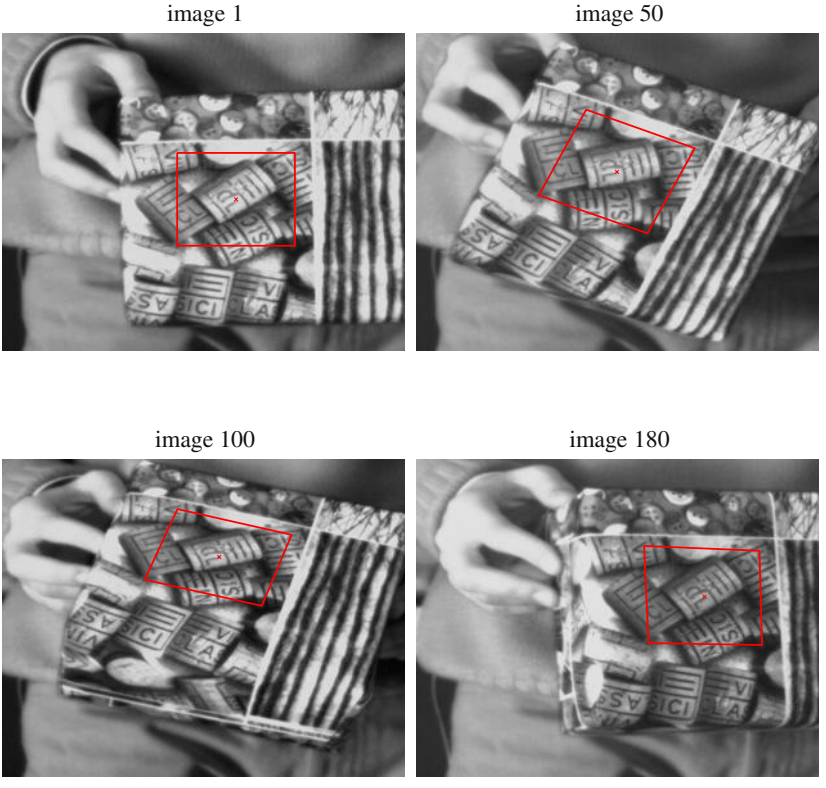


Fig. 8.5 Images from the Corkes sequence. The quadrilateral represents the tracked area. The visual tracking is correctly performed in real-time. However, the noise in the images and modeling errors affect the accuracy of the measured homographies

Figures 8.3 and 8.4 illustrate the instability of the critical points. In this simulation, the initial estimation error for the homography matrix is chosen at a critical point. The velocity estimate is again set to zero. Figure 8.3 shows that the critical point is unstable: a small noise allows the estimated homography to converge towards the measured homography. Figure 8.4 shows that the estimated velocity converges towards the true velocity.

8.5 Experiments with Real Data

In this section, we present results obtained with real data. In the first image the user selects a rectangular area of interest. The homographies that transform the area of interest in the current image are measured using the efficient second order minimization (ESM) visual tracking software[2] [1]. Figure 8.5 shows four images

[2] Available for download at http://esm.gforge.inria.fr.

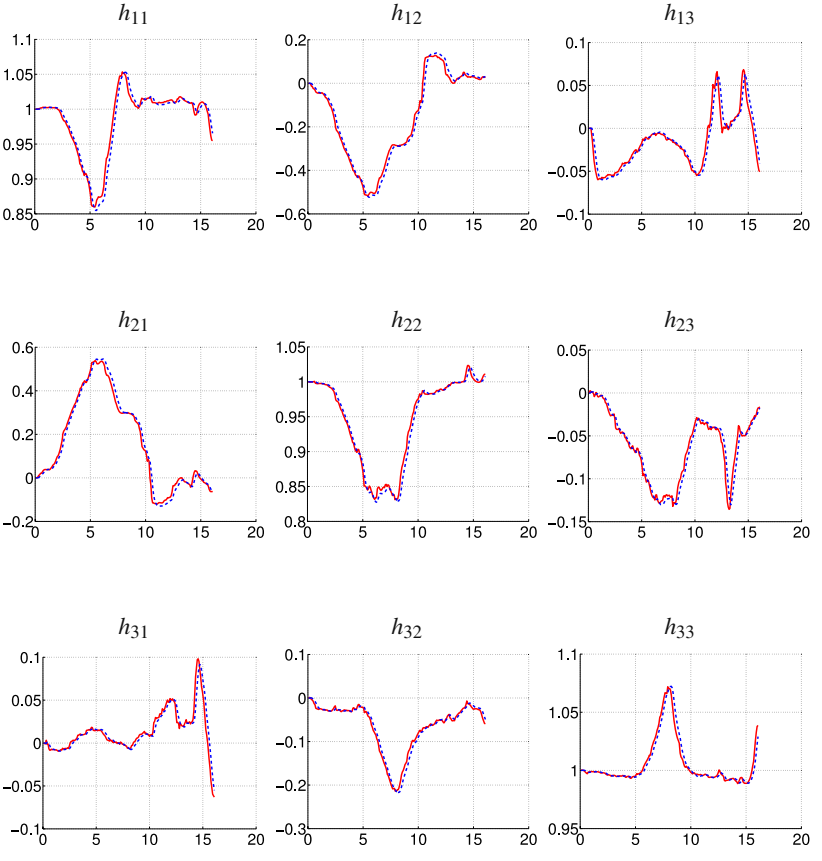


Fig. 8.6 Corkes sequence. Each plot represents an element of the (3×3) homography matrix. Plain line: the measured homography matrix H. Dashed line: the observed homography $\widehat{H}$

extracted from the sequence Corkes. The first image in the figure shows a rectangle containing the area of interest that must be tracked in all the images of the video sequence. For each image of the sequence, the output of the ESM visual tracking algorithm is the homography that encodes the transformation of each pixels of the rectangular area from the current to the first image.

The measured homographies are the input of the proposed nonlinear observer. In this experiment the gains were $k_H = 5$ and $k_A = 1$. The filtering effect of the observer on the estimated homography are visible in Figure 8.6.

In this experiment with real data, the velocity A is unknown and not constant. Nevertheless, the observer provides a smoothed estimation of the homography velocity, as illustrated on Figure 8.7.

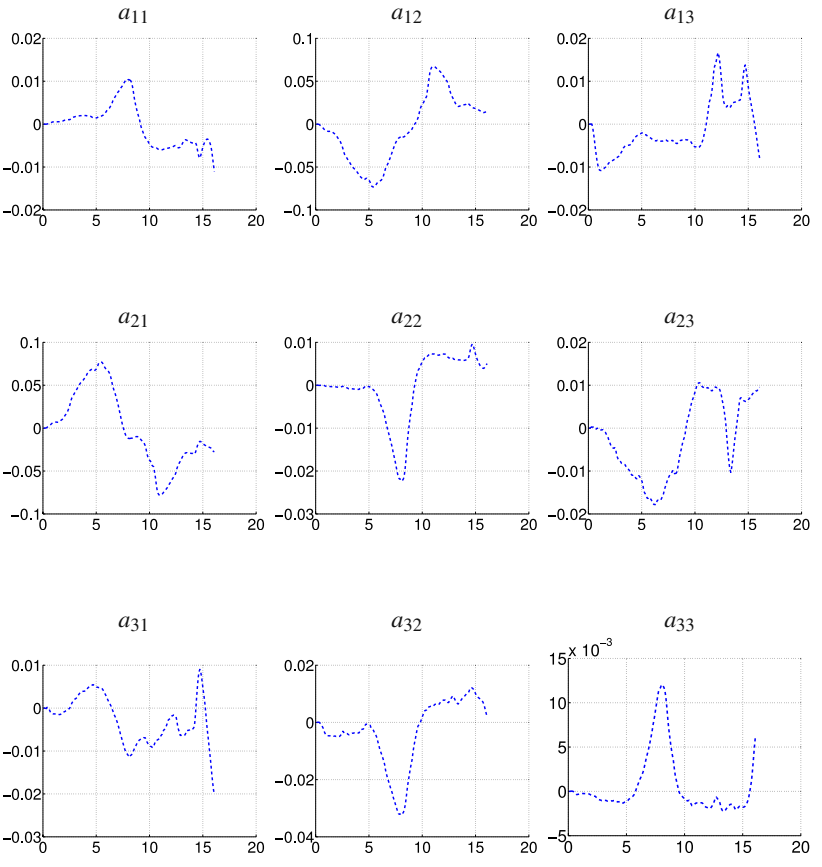


Fig. 8.7 Corkes sequence. Each plot represents an element of the observed homography velocity matrix $\widehat{A}$

8.6 Conclusion

In this chapter, we proposed an observer for the homographies defined on $SL(3)$ and their velocities defined on $\mathfrak{sl}(3)$. We proved that the observer is almost globally stable. We also proved that isolated critical points exist but that they are far from the equilibrium point and unstable. We performed several simulations with ground truth to validate the theoretical results. Experiments with real data show that the observer performs well even when the constant velocity assumption does not hold.

Acknowledgements. The authors gratefully acknowledge the contribution of INRIA, CNRS and ANU. This research was partly supported by the Australian Research Council through discovery grant DP0987411 and the PEGASE EU project.

References

[1] Benhimane, S., Malis, E.: Homography-based 2d visual tracking and servoing. International Journal of Robotic Research 26(7), 661–676 (2007)

[2] Benhimane, S., Malis, E., Rives, P., Azinheira, J.R.: Vision-based control for car platooning using homography decomposition. In: IEEE Conf. on Robotics and Automation, pp. 2173–2178 (2005)

[3] Bonnabel, S., Martin, P., Rouchon, P.: Non-linear observer on lie groups for left-invariant dynamics with right-left equivariant output. In: IFAC World Congress, pp. 8594–8598 (2008)

[4] Bonnabel, S., Martin, P., Rouchon, P.: Symmetry-preserving observers. IEEE Trans. on Automatic Control 53(11), 2514–2526 (2008)

[5] Bonnabel, S., Rouchon, P.: Invariant Observers. In: Bonnabel, S., Rouchon, P. (eds.) Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems. LNCIS, pp. 53–67. Springer, Heidelberg (2005)

[6] Chaumette, F., Hutchinson, S.: Visual servo control, part ii: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)

[7] Deguchi, K.: Optimal motion control for image-based visual servoing by decoupling translation and rotation. In: IEEE/RSJ Conf. on Intelligent Robots and Systems, pp. 705–711 (1998)

[8] Fang, Y., Dixon, W., Dawson, D., Chawda, P.: Homography-based visual servoing of wheeled mobile robots. IEEE Trans. on Systems, Man, and Cybernetics - Part B 35(5), 1041–1050 (2005)

[9] Faugeras, O., Lustman, F.: Motion and structure from motion in a piecewise planar environment. International Journal of Pattern Recognition and Artificial Intelligence 2(3), 485–508 (1988)

[10] Hamel, T., Mahony, R.: Attitude estimation on $SO(3)$ based on direct inertial measurements. In: IEEE Conf. on Robotics and Automation, pp. 2170–2175 (2006)

[11] Hashimoto, K.: Visual Servoing: Real Time Control of Robot manipulators based on visual sensory feedback. World Scientific Series in Robotics and Automated Systems, vol. 7. World Scientific Press, Singapore (1993)

[12] Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Trans. on Robotics and Automation 12(5), 651–670 (1996)

[13] Lageman, C., Mahony, R., Trumpf, J.: State observers for invariant dynamics on a lie group. In: Conf. on the Mathematical Theory of Networks and Systems (2008)

[14] Lageman, C., Mahony, R., Trumpf, J.: Gradient-like observers for invariant dynamics on a lie group. IEEE Trans. on Automatic Control (2009)

[15] Mahony, R., Hamel, T., Pflimlin, J.M.: Non-linear complementary filters on the special orthogonal group. IEEE Trans. on Automatic Control 53(5), 1203–1218 (2008)

[16] Malis, E., Chaumette, F.: Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. IEEE Trans. on Robotics and Automation 18(2), 176–186 (2002)

[17] Malis, E., Chaumette, F., Boudet, S.: 2 1/2 D visual servoing. IEEE Trans. on Robotics and Automation 15(2), 234–246 (1999)

[18] Malis, E., Vargas, M.: Deeper understanding of the homography decomposition for vision-based control. Research Report 6303, INRIA (2007)

[19] Martin, P., Salaün, E.: Invariant observers for attitude and heading estimation from low-cost inertial and magnetic sensors. In: IEEE Conf. on Decision and Control, pp. 1039–1045 (2007)

[20] Martin, P., Salaün, E.: An invariant observer for earth-velocity-aided attitude heading reference systems. In: IFAC World Congress, pp. 9857–9864 (2008)
[21] Samson, C., Le Borgne, M., Espiau, B.: Robot Control: the Task Function Approach. Oxford Engineering Science Series, vol. 22. World Scientific Press, Clarendon Press (1991)
[22] Suter, D., Hamel, T., Mahony, R.: Visual servo control using homography estimation for the stabilization of an x4-flyer. In: IEEE Conference on Decision and Control, pp. 2872–2877 (2002)
[23] Thienel, J., Sanner, R.M.: A coupled nonlinear spacecraft attitude controller and observer with an unknow constant gyro bias and gyro noise. IEEE Trans. on Automatic Control 48(11), 2011–2015 (2003)
[24] Vargas, M., Malis, E.: Visual servoing based on an analytical homography decomposition. In: IEEE Conf. on Decision and Control and European Control Conf., pp. 5379–5384 (2005)
[25] Wilson, W.J., Hulls, C.C.W., Bell, G.S.: Relative end-effector control using cartesian position-based visual servoing. IEEE Trans. on Robotics and Automation 12(5), 684–696 (1996)

Chapter 9
Image Measurement Errors in Visual Servoing: Estimating the Induced Positioning Error

Graziano Chesi, Yeung Sam Hung, and Ho Lam Yung

Abstract. The goal of a visual servo system is to position a robot end-effector by progressively adjusting its location so that some object features in the current image match the same features in a desired image previously recorded. However, this matching in the image domain cannot be ensured due to unavoidable presence of image measurement errors, and even when it is realized, there is no guarantee that the robot end-effector has reached the desired location since the available image measurements are corrupted. The aim of this chapter is to present a strategy for bounding the worst-case robot positioning error introduced by image measurement errors. In particular, two methods are described, which allow one to compute upper and lower bounds of this positioning error. Some examples illustrate the proposed methods with synthetic and real data.

9.1 Introduction

The teaching-by-showing approach is a fundamental technique for eye-in-hand visual servoing. In this approach, a robot end-effector mounting a camera is positioned at a location of interest, called desired location, from which some object features are visible. The view of the camera at this location, called desired view, is hence stored. Then, the camera is moved to another location of the scene from which the same object features are visible. The target is to steer the camera from its current location to the desired location by exploiting as feedback information the view of the camera in the current location, called current view, and the desired view previously stored.

Various methods have been proposed in the literature for the teaching-by-showing approach, see for instance [21, 19, 4, 5]. Among these, there are basically two pioneering methods: image-based visual servoing (IBVS), see *e.g.* [20, 3], and

Graziano Chesi, Yeung Sam Hung, and Ho Lam Yung
Department of Electrical and Electronic Engineering, University of Hong Kong,
Pokfulam Road, Hong Kong
e-mail: {chesi,yshung,h0440266}@eee.hku.hk

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 151–167.
springerlink.com

position-based visual servoing (PBVS), see *e.g.* [30]. In IBVS, the feedback error is defined in the image domain, in particular as the difference between the current and the desired vectors of coordinates of the object features. In PBVS, the feedback error is defined in the 3D space, in particular as the camera pose between the current and the desired locations. Starting from these two methods, several others have been derived, for example proposing the use of a feedback error defined in both image domain and 3D space (see *e.g.* [25]), partition of the degrees of freedoms (see *e.g.* [16]), global motion plan via navigation functions (see *e.g.* [17]), control invariant with respect to intrinsic parameters (see *e.g.* [24, 28]), use of complex image features via image moments (see *e.g.* [29]), switching strategies for ensuring the visibility constraint (see *e.g.* [9, 18]), generation of circular-like trajectories for minimizing the trajectory length and achieving global convergence (see *e.g.* [14]), and path-planning methods that take constraints into account (see *e.g.* [26, 27, 1, 10, 6]).

In all these methods the goal condition is that the image features in the current view have to match their corresponding ones in the desired view. However, the fulfillment of this condition can never be ensured due to the presence of image noise, which leads to unavoidable image measurement errors. Moreover, even when fulfilled, this condition can never guarantee that the robot end-effector has reached the sought desired location since the available measurements are corrupted. Hence, this problem is of fundamental importance in visual servoing. See also [23, 22] where the effect of image noise on the control laws of PBVS, 2 1/2 D visual servoing, and IBVS, has been investigated.

This chapter addresses the estimation of the worst-case robot positioning error introduced by image measurement errors. Specifically, a strategy for computing an estimate of the set of admissible values of this worst-case error is proposed. This estimate is obtained by solving optimizations over polynomials with linear matrix inequalities (LMIs) and barrier functions, which provide upper and lower bounds to the sought worst-case robot positioning error. These optimizations are built by introducing suitable parametrizations of the camera frame and by adopting the square matrix representation (SMR) of polynomials. Some examples with synthetic and real data illustrate the application of the proposed strategy. This chapter extends our previous results in [11, 15].

The organization of the chapter is as follows. Section 9.2 introduces the problem formulation and some preliminaries. Section 9.3 describes the computation of the upper and the lower bounds. Section 9.4 presents some illustrative examples. Lastly, Section 9.5 concludes the chapter with some final remarks.

9.2 Preliminaries

In this section we describe the notation adopted throughout the chapter, we provide the formulation of the problem, and we briefly explain how polynomials can be represented by using the SMR.

9.2.1 Notation and Problem Statement

The notation used in this chapter is as follows:

- $\mathbb{R}$: real number set;
- $SO(3)$: set of all 3×3 rotation matrices;
- $\mathbf{0}_n$: null $n \times 1$ vector;
- $\mathbf{0}_{m \times n}$: null $m \times n$ matrix;
- $\mathbf{I}_n$: $n \times n$ identity matrix;
- $\mathbf{e}_i$: i-th column of $\mathbf{I}_3$;
- $\|\mathbf{X}\|$: Euclidean norm;
- $\|\mathbf{X}\|_\infty$: infinity norm;
- $\mathbf{X}^T$: transpose; and
- s.t.: subject to.

In an eye-in-hand visual servo system, such as a robotic manipulator or a mobile platform with a camera mounted on an end-effector, the goal consists of controlling the robot so that the end-effector reaches a desired location by exploiting as feedback information the image projections of some object features. In particular, the goal is assumed to be achieved when the object features in the current view match the corresponding ones in the desired view, which have previously been recorded.

Image noise (for instance due to image quantization, lighting, features extraction, *etc*) unavoidably affects the image measurements, *i.e.* the estimate of the position of the object features in the image. This means that the matching between object features in the current view and in the desired view can never be ensured as image measurement errors are nondeterministic. Moreover, even when this matching is realized, there is no guarantee that the robot end-effector has reached the desired location since the available image measurements are corrupted.

This chapter addresses the problem of bounding the worst-case robot positioning error introduced by image measurement errors, which depends on the level of image noise, camera parameters, and object features.

9.2.2 Mathematical Formulation of the Problem

Let F^{abs} be an absolute frame in the 3D space. We denote with $F = (\mathbf{O}, \mathbf{c})$ the current camera frame expressed with respect to F^{abs}, where $\mathbf{O} \in SO(3)$ is a rotation matrix defining the orientation, and $\mathbf{c} \in \mathbb{R}^3$ is a vector defining the translation. Similarly, we denote with $F^* = (\mathbf{O}^*, \mathbf{c}^*)$ the desired camera frame.

Let $\mathbf{q}_1, \ldots, \mathbf{q}_N \in \mathbb{R}^3$ be a set of 3D points expressed with respect to F^{abs}. The i-th 3D point $\mathbf{q}_i$ projects onto F at the point $\mathbf{p}_i = (p_{i,1}, p_{i,2}, 1)^T \in \mathbb{R}^3$ given by

$$d_i \mathbf{p}_i = \mathbf{A}\mathbf{O}^T (\mathbf{q}_i - \mathbf{c}) \tag{9.1}$$

where d_i is the depth of the point with respect to F, and $\mathbf{A} \in \mathbb{R}^{3\times3}$ is the upper-triangular matrix containing the camera intrinsic parameters according to

$$\mathbf{A} = \begin{pmatrix} f_x & s & u_x \\ 0 & f_y & u_y \\ 0 & 0 & 1 \end{pmatrix}, \tag{9.2}$$

$f_x, f_y \in \mathbb{R}$ being the focal lengths, $u_x, u_y \in \mathbb{R}$ the coordinates of the principal point, and $s \in \mathbb{R}$ the aspect ratio. Similarly, $\mathbf{q}_i$ projects onto F^* at the point $\mathbf{p}_i^* = (p_{i,1}^*, p_{i,2}^*, 1)^T \in \mathbb{R}^3$ given by

$$d_i^* \mathbf{p}_i^* = \mathbf{A}\mathbf{O}^{*T} (\mathbf{q}_i - \mathbf{c}^*) \tag{9.3}$$

where d_i^* is the depth of the point with respect to F^*. The camera pose between F and F^* is described by the pair

$$(\mathbf{R}, \mathbf{t}) \in SO(3) \times \mathbb{R}^3 \tag{9.4}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the rotational and translational components expressed with respect to F^* and given by

$$\begin{aligned} \mathbf{R} &= \mathbf{O}^{*T}\mathbf{O} \\ \mathbf{t} &= \frac{\mathbf{O}^{*T}(\mathbf{c} - \mathbf{c}^*)}{\|\mathbf{O}^{*T}(\mathbf{c} - \mathbf{c}^*)\|} \end{aligned} \tag{9.5}$$

($\mathbf{t}$ is normalized because, by exploiting only the image projections of the points $\mathbf{q}_1, \ldots, \mathbf{q}_N$ and the matrix $\mathbf{A}$, the translation can be recovered only up to a scale factor). Let $\mathbf{p}, \mathbf{p}^* \in \mathbb{R}^{2N}$ be the vectors defined as

$$\mathbf{p} = \begin{pmatrix} p_{1,1} \\ p_{1,2} \\ \vdots \\ p_{N,1} \\ p_{N,2} \end{pmatrix}, \quad \mathbf{p}^* = \begin{pmatrix} p_{1,1}^* \\ p_{1,2}^* \\ \vdots \\ p_{N,1}^* \\ p_{N,2}^* \end{pmatrix}.$$

The goal condition of an eye-in-hand visual servo system can be expressed as

$$\|\mathbf{p} - \mathbf{p}^*\|_\infty \leq \varepsilon \tag{9.6}$$

where $\varepsilon \in \mathbb{R}$ is a threshold chosen to limit the distance between $\mathbf{p}$ and $\mathbf{p}^*$ (for example, via the infinity norm).

This chapter addresses the computation of upper and lower bounds of the worst-case robot positioning error introduced by image measurement errors through the goal condition (9.6). In particular, we consider the worst-case rotational error

$$s_r(\varepsilon) = \sup_{\mathbf{R},\mathbf{t}} \theta \ \text{ s.t. } \ \|\mathbf{p} - \mathbf{p}^*\|_\infty \leq \varepsilon \tag{9.7}$$

where $\theta \in [0, \pi]$ is the angle in the representation of $\mathbf{R}$ via exponential coordinates, *i.e.*

$$\mathbf{R} = e^{[\theta \mathbf{u}]_\times} \tag{9.8}$$

for some $\mathbf{u} \in \mathbb{R}^3$ with $\|\mathbf{u}\| = 1$. Also, we consider the worst-case translational error

$$s_t(\varepsilon) = \sup_{\mathbf{R},\mathbf{t}} \|\mathbf{t}\| \ \text{ s.t. } \ \|\mathbf{p} - \mathbf{p}^*\|_\infty \leq \varepsilon. \tag{9.9}$$

In the sequel we will consider without loss of generality that F^* coincides with F^{abs}.

9.2.3 SMR of Polynomials

The approach proposed in this chapter is based on the SMR of polynomials introduced in [13] to solve optimization problems over polynomials via LMI techniques. Specifically, let $y(\mathbf{x})$ be a polynomial of degree $2m$ in the variable $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, *i.e.*

$$y(\mathbf{x}) = \sum_{\substack{i_1 + \ldots + i_n \leq 2m \\ i_1 \geq 0, \ldots, i_n \geq 0}} y_{i_1,\ldots,i_n} x_1^{i_1} \cdots x_n^{i_n}$$

for some coefficients $y_{i_1,\ldots,i_n} \in \mathbb{R}$. According to the SMR, $y(\mathbf{x})$ can be expressed as

$$y(\mathbf{x}) = \mathbf{v}(\mathbf{x})^T \left(\mathbf{Y} + \mathbf{L}(\alpha)\right) \mathbf{v}(\mathbf{x})$$

where $\mathbf{v}(\mathbf{x})$ is any vector containing a base for the polynomials of degree m in $\mathbf{x}$, and hence can be simply chosen such that each of its entry is a monomial of degree less than or equal to m in $\mathbf{x}$, for example

$$\mathbf{v}(\mathbf{x}) = (1, x_1, \ldots, x_n, x_1^2, x_1 x_2, \ldots, x_n^m)^T.$$

The matrix $\mathbf{Y}$ is any symmetric matrix such that

$$y(\mathbf{x}) = \mathbf{v}(\mathbf{x})^T \mathbf{Y} \mathbf{v}(\mathbf{x})$$

which can be simply obtained via trivial coefficient comparisons. The vector α is a vector of free parameters, and the matrix function $\mathbf{L}(\alpha)$ is a linear parametrization of the linear set

$$\mathscr{L} = \{\mathbf{L} = \mathbf{L}^T : \ \mathbf{v}(\mathbf{x})^T \mathbf{L} \mathbf{v}(\mathbf{x}) = 0 \quad \forall \mathbf{x}\}$$

which can be computed through standard linear algebra techniques for parameterizing linear spaces.

The matrices $\mathbf{Y}$ and $\mathbf{Y} + \mathbf{L}(\alpha)$ are known as SMR matrix and complete SMR matrix of $y(\mathbf{x})$. The length of $\mathbf{v}(\mathbf{x})$ is given by

$$d_1(n,m) = \frac{(n+m)!}{n!m!}$$

while the length of α (*i.e.*, the dimension of $\mathscr{L}$) is

$$d_2(n,m) = \frac{1}{2} d_1(n,m)(d_1(n,m)+1) - d_1(n,2m).$$

The SMR is useful because it allows one to establish positivity of polynomials by solving a convex optimization problem with LMIs. Indeed, $y(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$ if $y(\mathbf{x})$ is a sum of squares of polynomials (SOS), *i.e.* if there exists $y_1(\mathbf{x}), \ldots, y_k(\mathbf{x})$ such that

$$y(\mathbf{x}) = \sum_{i=1}^{k} y_i(\mathbf{x})^2.$$

It turns out that $y(\mathbf{x})$ is SOS if and only if there exists α such that

$$\mathbf{Y} + \mathbf{L}(\alpha) \geq 0$$

which is an LMI. As explained for example in [2], the feasible set of an LMI is convex, and to establish whether such a set is nonempty amounts to solving a convex optimization problem.

See *e.g.* [13, 7, 8] for details on the SMR and for algorithms for constructing SMR matrices.

9.3 Computation of the Bounds

Let us denote with $\hat{\mathbf{p}}^*$ and $\hat{\mathbf{p}}$ the available estimates of $\mathbf{p}^*$ and $\mathbf{p}$ corrupted by image noise. According to (9.6), the goal condition in visual servoing is

$$\|\hat{\mathbf{p}} - \hat{\mathbf{p}}^*\|_\infty \leq \varepsilon. \tag{9.10}$$

The estimates $\hat{\mathbf{p}}^*$ and $\hat{\mathbf{p}}$ are related to $\mathbf{p}^*$ and $\mathbf{p}$ by

$$\begin{cases} \hat{\mathbf{p}} = \mathbf{p} + \mathbf{n} \\ \hat{\mathbf{p}}^* = \mathbf{p}^* + \mathbf{n}^* \end{cases}$$

where $\mathbf{n}, \mathbf{n}^* \in \mathbb{R}^{2N}$ are vectors containing position errors due to image noise. Suppose that $\mathbf{n}$ and $\mathbf{n}^*$ are bounded by

$$\begin{aligned} \|\mathbf{n}\|_\infty &\leq \zeta \\ \|\mathbf{n}^*\|_\infty &\leq \zeta \end{aligned}$$

where $\zeta \in \mathbb{R}$ is a bound of the position errors in both current and desired views. One has:

$$\begin{aligned} \|\mathbf{p} - \mathbf{p}^*\|_\infty &= \|\hat{\mathbf{p}} - \hat{\mathbf{p}}^* - \mathbf{n} + \mathbf{n}^*\|_\infty \\ &\leq \|\hat{\mathbf{p}} - \hat{\mathbf{p}}^*\|_\infty + \|\mathbf{n}\|_\infty + \|\mathbf{n}^*\|_\infty \\ &\leq \|\hat{\mathbf{p}} - \hat{\mathbf{p}}^*\|_\infty + 2\zeta. \end{aligned}$$

This implies that the condition (9.10) ensures only

$$\|\mathbf{p} - \mathbf{p}^*\|_\infty \leq \varepsilon + 2\zeta.$$

Hence, one cannot guarantee that the real image error $\|\mathbf{p} - \mathbf{p}^*\|_\infty$ converges to a value smaller than 2ζ. This clearly motivates the investigation of the robot

positioning error introduced by image measurement errors. For this reason, from now on we will consider the computation of the errors $s_r(\delta)$ and $s_t(\delta)$ where δ, defined as

$$\delta = \varepsilon + 2\zeta, \tag{9.11}$$

represents the total image error.

Before proceeding, let us parameterize the rotation matrix through the Cayley parameter. Let us define the function $\Gamma : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$

$$\Gamma(\mathbf{a}) = \left(\mathbf{I}_3 - [\mathbf{a}]_\times\right)^{-1}\left(\mathbf{I}_3 + [\mathbf{a}]_\times\right) \tag{9.12}$$

where $\mathbf{a} = (a_1, a_2, a_3)^T \in \mathbb{R}^3$ is called Cayley parameter. It turns out that this parametrization satisfies the following properties:

1. $\Gamma(\mathbf{a}) \in SO(3)$ for all $\mathbf{a} \in \mathbb{R}^3$;
2. for any $\mathbf{R} \in SO(3)$ such that θ in (9.8) satisfies $\theta < \pi$, there exists $\mathbf{a} \in \mathbb{R}^3$ such that $\mathbf{R} = \Gamma(\mathbf{a})$.

Moreover, (9.12) can be rewritten as

$$\Gamma(\mathbf{a}) = \frac{\Omega(\mathbf{a})}{1 + \|\mathbf{a}\|^2} \tag{9.13}$$

where $\Omega(\mathbf{a}) : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ is the matrix quadratic polynomial given by

$$\Omega(\mathbf{a}) = \begin{pmatrix} a_1^2 - a_2^2 - a_3^2 + 1 & 2(a_1a_2 - a_3) & 2(a_1a_3 + a_2) \\ 2(a_1a_2 + a_3) & -a_1^2 + a_2^2 - a_3^2 + 1 & 2(a_2a_3 - a_1) \\ 2(a_1a_3 - a_2) & 2(a_2a_3 + a_1) & -a_1^2 - a_2^2 + a_3^2 + 1 \end{pmatrix}. \tag{9.14}$$

9.3.1 Upper Bounds

Let us consider first the computation of upper bounds of $s_r(\delta)$ and $s_t(\delta)$ in (9.7)–(9.9). We will show that this step can be solved by exploiting convex optimization. Indeed, consider the constraint $\|\mathbf{p} - \mathbf{p}^*\|_\infty < \delta$ in the computation of $s_r(\delta)$ and $s_t(\delta)$. From (9.1), (9.3) and (9.5) it follows that for the i-th point we can write

$$\mathbf{p}_i - \mathbf{p}_i^* = \mathbf{A}\frac{\Omega(\mathbf{a})^T\mathbf{q}_i - \Omega(\mathbf{a})^T\mathbf{t}}{\mathbf{e}_3^T\left(\Omega(\mathbf{a})^T\mathbf{q}_i - \Omega(\mathbf{a})^T\mathbf{t}\right)} - \mathbf{A}\frac{\mathbf{q}_i}{\mathbf{e}_3^T\mathbf{q}_i} \tag{9.15}$$

where it has been taken into account that $\mathbf{q}_i$ is expressed with respect to the desired camera frame F^*, which coincides with the absolute frame F^{abs}. Hence, we have that

$$\|\mathbf{p}_i - \mathbf{p}_i^*\|_\infty \le \delta$$

if and only if

$$\begin{cases} |f_{i,3}g_{i,1}(\mathbf{a},\mathbf{t}) - f_{i,1}g_{i,3}(\mathbf{a},\mathbf{t})| \le \delta f_{i,3}g_{i,3}(\mathbf{a},\mathbf{t}) \\ |f_{i,3}g_{i,2}(\mathbf{a},\mathbf{t}) - f_{i,2}g_{i,3}(\mathbf{a},\mathbf{t})| \le \delta f_{i,3}g_{i,3}(\mathbf{a},\mathbf{t}) \\ g_{i,3}(\mathbf{a},\mathbf{t}) > 0 \end{cases} \tag{9.16}$$

where $f_{i,j} \in \mathbb{R}$ is a constant and $g_{i,j}(\mathbf{a},\mathbf{t})$ is a polynomial given by

$$\begin{aligned} f_{i,j} &= \mathbf{e}_j^T \mathbf{A} \mathbf{q}_i \\ g_{i,j}(\mathbf{a},\mathbf{t}) &= \mathbf{e}_j^T \mathbf{A} \left(\Omega(\mathbf{a})^T \mathbf{q}_i - \Omega(\mathbf{a})^T \mathbf{t} \right). \end{aligned}$$

This means that the constraint $\|\mathbf{p}-\mathbf{p}^*\|_\infty < \delta$ can be compactly expressed by defining the polynomials

$$h_{i,j,k}(\mathbf{a},\mathbf{t}) = (-1)^k \left(f_{i,3} g_{i,j}(\mathbf{a},\mathbf{t}) - f_{i,j} g_{i,3}(\mathbf{a},\mathbf{t}) \right) + \delta f_{i,3} g_{i,3}(\mathbf{a},\mathbf{t}). \tag{9.17}$$

Indeed, $\|\mathbf{p}-\mathbf{p}^*\|_\infty \leq \delta$ if and only if

$$h_{i,j,k}(\mathbf{a},\mathbf{t}) \geq 0 \ \ \forall (i,j,k) \in \mathscr{I} \tag{9.18}$$

where $\mathscr{I}$ is the set

$$\begin{aligned} \mathscr{I} = & \{(i,j,k) : \ i = 1,\ldots,N, \ j = 1,2, \ k = 1,2\} \\ & \bigcup \ \{(i,j,k) : \ i = 1,\ldots,N, \ j = 3, \ k = 1\}. \end{aligned} \tag{9.19}$$

In fact, let us observe that the first two constraints in (9.16) are recovered by the first set on the right hand side of (9.19), while the third constraint is recovered by the second set on the right hand side of (9.19) since $h_{i,3,1}(\mathbf{a},\mathbf{t}) = \delta f_{i,3} g_{i,3}(\mathbf{a},\mathbf{t})$. Therefore, $s_r(\delta)$ and $s_t(\delta)$ can be rewritten as

$$\begin{aligned} s_r(\delta) &= \sup_{\mathbf{a},\mathbf{t}} \ \theta \ \text{ s.t. (9.18)} \\ s_t(\delta) &= \sup_{\mathbf{a},\mathbf{t}} \ \|\mathbf{t}\| \ \text{ s.t. (9.18).} \end{aligned} \tag{9.20}$$

Let us observe that $g_{i,j}(\mathbf{a},\mathbf{t})$ is a cubic polynomial in the variables $\mathbf{a}$ and $\mathbf{t}$ due to the term $\Omega(\mathbf{a})^T \mathbf{t}$. In order to lower the degree of the polynomials in the optimization problems in (9.20), let us define the new variable $\mathbf{z} \in \mathbb{R}^3$ as

$$\mathbf{z} = \Omega(\mathbf{a})^T \mathbf{t}.$$

Let us observe that $\mathbf{t}$ can be recovered from $\mathbf{z}$ since $\Omega(\mathbf{a})^T$ is always invertible, in particular

$$\left(\Omega(\mathbf{a})^T \right)^{-1} = \frac{\Gamma(\mathbf{a})}{1+\|\mathbf{a}\|^2}.$$

Hence, let us define

$$\begin{aligned} \hat{g}_{i,j}(\mathbf{a},\mathbf{z}) &= \mathbf{e}_j^T \mathbf{A} \left(\Omega(\mathbf{a})^T \mathbf{q}_i - \mathbf{z} \right) \\ \hat{h}_{i,j,k}(\mathbf{a},\mathbf{z}) &= (-1)^k \left(f_{i,3} \hat{g}_{i,j}(\mathbf{a},\mathbf{z}) - f_{i,j} \hat{g}_{i,3}(\mathbf{a},\mathbf{z}) \right) + \delta f_{i,3} \hat{g}_{i,3}(\mathbf{a},\mathbf{z}). \end{aligned}$$

It follows that (9.18) holds if and only if

$$\hat{h}_{i,j,k}(\mathbf{a},\mathbf{z}) \geq 0 \ \ \forall (i,j,k) \in \mathscr{I}.$$

Moreover, the degree of $\hat{h}_{i,j,k}(\mathbf{a},\mathbf{z})$ is 2, which is lower than that of $h_{i,j,k}(\mathbf{a},\mathbf{t})$ which is equal to 3.

The next step consists of solving (9.20) via convex optimization. To this end, let us introduce the polynomials

$$\begin{aligned} b_r(\mathbf{a},\mathbf{z}) &= \gamma - \|\mathbf{a}\|^2 - \textstyle\sum_{(i,j,k)\in\mathscr{I}} u_{i,j,k}(\mathbf{a},\mathbf{z})\hat{h}_{i,j,k}(\mathbf{a},\mathbf{z}) \\ b_t(\mathbf{a},\mathbf{z}) &= (1+\|\mathbf{a}\|^2)^2\gamma - \|\mathbf{z}\|^2 - \textstyle\sum_{(i,j,k)\in\mathscr{I}} u_{i,j,k}(\mathbf{a},\mathbf{z})\hat{h}_{i,j,k}(\mathbf{a},\mathbf{z}) \end{aligned} \tag{9.21}$$

where $u_{i,j,k}(\mathbf{a},\mathbf{z}) \in \mathbb{R}$ are auxiliary polynomials and $\gamma \in \mathbb{R}$ is an auxiliary scalar to be determined. Let us exploit the SMR of polynomials described in Section 9.2.3. Let $\mathbf{v}_b(\mathbf{a},\mathbf{z})$ be a vector containing any base for the polynomials $b_r(\mathbf{a},\mathbf{z})$ and $b_t(\mathbf{a},\mathbf{z})$, and let $\mathbf{v}_u(\mathbf{a},\mathbf{z})$ be a similar vector for the polynomials $u_{i,j,k}(\mathbf{a},\mathbf{z})$. Then, the polynomials $b_r(\mathbf{a},\mathbf{z}), b_t(\mathbf{a},\mathbf{z})$ and $u_{i,j,k}(\mathbf{a},\mathbf{z})$ can be expressed as

$$\begin{cases} b_r(\mathbf{a},\mathbf{z}) = \mathbf{v}_b(\mathbf{a},\mathbf{z})^T \mathbf{B}_r \mathbf{v}_b(\mathbf{a},\mathbf{z}) \\ b_t(\mathbf{a},\mathbf{z}) = \mathbf{v}_b(\mathbf{a},\mathbf{z})^T \mathbf{B}_t \mathbf{v}_b(\mathbf{a},\mathbf{z}) \\ u_{i,j,k}(\mathbf{a},\mathbf{z}) = \mathbf{v}_u(\mathbf{a},\mathbf{z})^T \mathbf{U}_{i,j,k} \mathbf{v}_u(\mathbf{a},\mathbf{z}) \end{cases} \tag{9.22}$$

where $\mathbf{B}_r$, $\mathbf{B}_t$ and $\mathbf{U}_{i,j,k}$ are any symmetric matrices of suitable dimensions satisfying (9.22). Let $\mathbf{L}(\alpha)$ be any linear parametrization of the linear set

$$\mathscr{L} = \left\{ \mathbf{L} = \mathbf{L}^T : \ \mathbf{v}_b(\mathbf{a},\mathbf{z})^T \mathbf{L} \mathbf{v}_b(\mathbf{a},\mathbf{z}) = 0 \quad \forall \mathbf{a},\mathbf{z} \right\}$$

where α is a free vector, and let us define the optimization problems

$$\begin{aligned} \gamma_r &= \inf_{\gamma,\alpha,\mathbf{U}_{i,j,k}} \gamma \ \text{ s.t. } \begin{cases} \mathbf{B}_r + \mathbf{L}(\alpha) \geq 0 \\ \mathbf{U}_{i,j,k} \geq 0 \ \ \forall (i,j,k) \in \mathscr{I} \end{cases} \\ \gamma_t &= \inf_{\gamma,\alpha,\mathbf{U}_{i,j,k}} \gamma \ \text{ s.t. } \begin{cases} \mathbf{B}_t + \mathbf{L}(\alpha) \geq 0 \\ \mathbf{U}_{i,j,k} \geq 0 \ \ \forall (i,j,k) \in \mathscr{I}. \end{cases} \end{aligned} \tag{9.23}$$

These optimization problems are convex since the cost functions are linear and the feasible sets are convex being the feasible sets of LMIs. In particular, these problems belong to the class of eigenvalue problems (EVPs), also known as semidefinite programming [2].

Let us observe that the constraints in (9.23) ensure that

$$\left.\begin{array}{l} b_r(\mathbf{a},\mathbf{z}) \geq 0 \\ b_t(\mathbf{a},\mathbf{z}) \geq 0 \\ u_{i,j,k}(\mathbf{a},\mathbf{z}) \geq 0 \end{array}\right\} \ \forall \mathbf{a},\mathbf{z}$$

from which one obtains

$$\left.\begin{array}{l} \gamma_r \geq \|\mathbf{a}\|^2 \\ \gamma_t \geq \dfrac{\|\mathbf{z}\|^2}{(1+\|\mathbf{a}\|^2)^2} \end{array}\right\} \ \forall \mathbf{a},\mathbf{z} : \ h_{i,j,k}(\mathbf{a},\mathbf{z}) \geq 0 \ \forall (i,j,k) \in \mathscr{I}.$$

Moreover, for the rotation one has that (see for instance [12])

$$\|\mathbf{a}\|^2 = \left(\tan\frac{\theta}{2}\right)^2$$

and for the translation

$$\|\mathbf{z}\|^2 = (1+\|\mathbf{a}\|^2)^2\|\mathbf{t}\|^2.$$

Consequently, an upper bound of $s_r(\delta)$ is given by

$$s_r^+(\delta) = 2\arctan\sqrt{\gamma_r} \tag{9.24}$$

and an upper bound of $s_t(\delta)$ is given by

$$s_t^+(\delta) = \sqrt{\gamma_t}. \tag{9.25}$$

9.3.2 Lower Bounds

In Section 9.3.1 we have derived upper bounds of the worst-case errors $s_r(\delta)$ and $s_t(\delta)$. In this section we consider the computation of lower bounds of these errors. The idea is to generate a sequence of camera poses $(\mathbf{R},\mathbf{t})$ such that:

1. the condition $\|\mathbf{p}-\mathbf{p}^*\|_\infty \le \delta$ holds true for all values of the sequence, *i.e.* each camera pose $(\mathbf{R},\mathbf{t})$ in the sequence determines an admissible image error and hence a lower bound of the sought worst-case error;
2. the sequence approaches the sought worst-case error.

Let us start by defining the functions

$$\psi_r = \begin{cases} \mu(\Omega(\mathbf{a})) & \text{if } w(\mathbf{a},\mathbf{t}) \ge 0 \\ 0 & \text{otherwise} \end{cases}$$
$$\psi_t = \begin{cases} \nu(\mathbf{t}) & \text{if } w(\mathbf{a},\mathbf{t}) \ge 0 \\ 0 & \text{otherwise} \end{cases}$$

where $w(\mathbf{a},\mathbf{t})$ is given by

$$w(\mathbf{a},\mathbf{t}) = \delta - \|\mathbf{p}-\mathbf{p}^*\|_\infty. \tag{9.26}$$

Let us observe that $w(\mathbf{a},\mathbf{t})$ is a barrier function, in particular it becomes negative whenever the parameters $\mathbf{a}$ and $\mathbf{t}$ are not admissible, *i.e.* $\|\mathbf{p}-\mathbf{p}^*\|_\infty > \delta$. Then, lower bounds of $s_r(\delta)$ and $s_t(\delta)$ can be obtained as

$$\begin{aligned} s_r^-(\delta) &= \sup_{\mathbf{a},\mathbf{t}} \ \psi_r \\ s_t^-(\delta) &= \sup_{\mathbf{a},\mathbf{t}} \ \psi_t. \end{aligned} \tag{9.27}$$

The optimization problems in (9.27) can be solved by using various techniques, such as the simplex algorithm which can handle also nondifferentiable cost functions. The found solutions may be nonoptimal since these functions are nonconvex, nevertheless it is guaranteed that any found solution is a lower bound of the sought worst-case error.

Lastly, let us observe that an initialization for the optimization problems in (9.27) is simply given by $(\mathbf{0}_3, \mathbf{0}_3)$, which is admissible for any possible δ. Indeed:

$$\begin{aligned} w(\mathbf{0}_3, \mathbf{0}_3) &= \delta - \|\mathbf{p}^* - \mathbf{p}^*\|_\infty \\ &= \delta \\ &\geq 0. \end{aligned}$$

9.4 Examples

In this section we present some examples of the proposed approach. The upper bounds $s_r^+(\delta)$ and $s_t^+(\delta)$ in (9.24)–(9.25) and the lower bounds $s_r^-(\delta)$ and $s_t^-(\delta)$ in (9.27) have been computed by using Matlab and the toolbox SeDuMi.

9.4.1 Example 1

Let us consider the situation shown in Figure 9.1(a) where a camera is observing four dice. The chosen object points are the centers of the ten large dots. The screen size is 640×480 pixels, and the intrinsic parameters matrix is

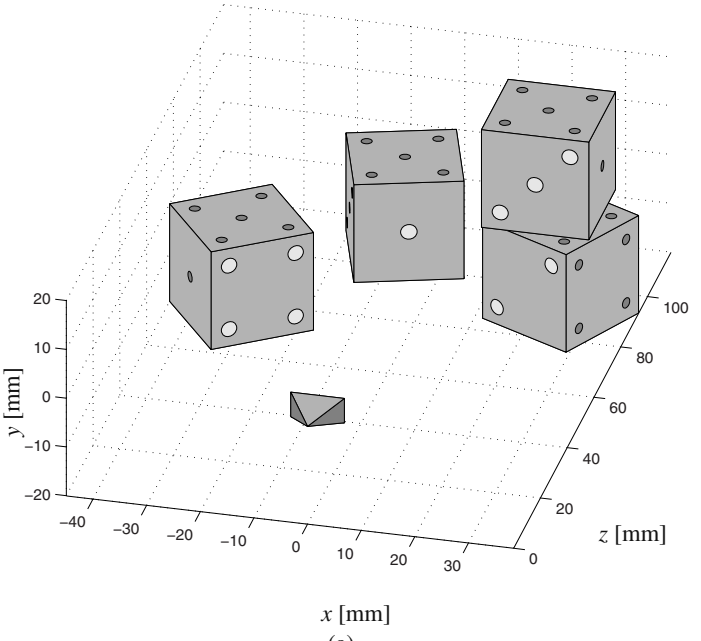
$$\mathbf{A} = \begin{pmatrix} 500 & 0 & 320 \\ 0 & 500 & 240 \\ 0 & 0 & 1 \end{pmatrix}.$$

Figure 9.1(b) shows the corresponding camera view. The problem is to estimate the worst-case location error introduced by image points matching, *i.e.* the errors $s_r(\delta)$ and $s_t(\delta)$ in (9.20) where δ represents the total image error in (9.11).

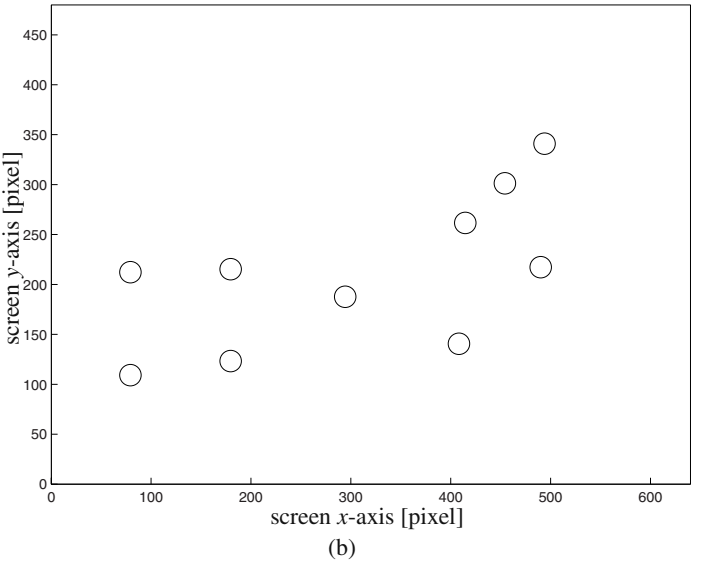
We compute the upper bounds $s_r^+(\delta)$ and $s_t^+(\delta)$ in (9.24)–(9.25) and the lower bounds $s_r^-(\delta)$ and $s_t^-(\delta)$ in (9.27) for some values of δ. We find the values shown in Table 9.1.

Table 9.1 Example 1: upper and lower bounds of the worst-case location errors for the object points in Figure 9.1

δ [pixels]	$s_r^-(\delta)$ [deg]	$s_r^+(\delta)$ [deg]	$s_t^-(\delta)$ [mm]	$s_t^+(\delta)$ [mm]
0.5	0.138	0.725	0.146	1.644
1.0	0.960	1.452	1.011	3.297
1.5	1.469	2.183	1.510	4.964
2.0	1.962	2.916	2.106	6.650

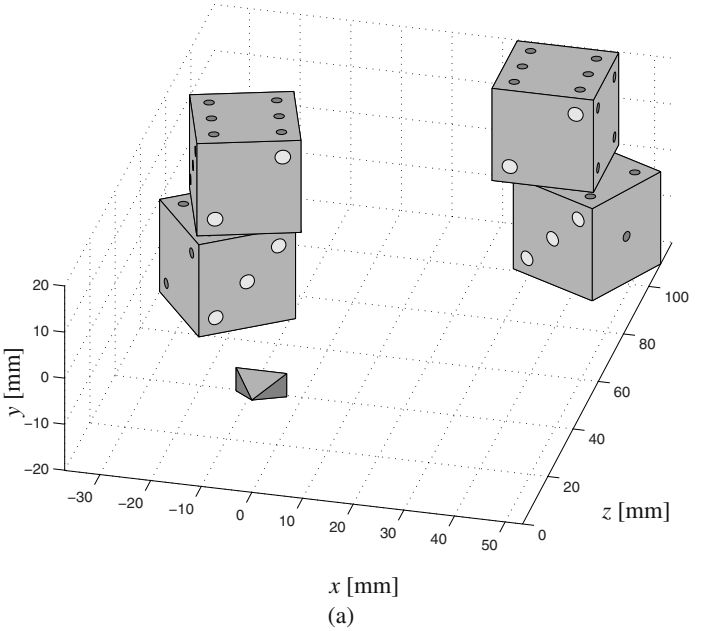


(a)

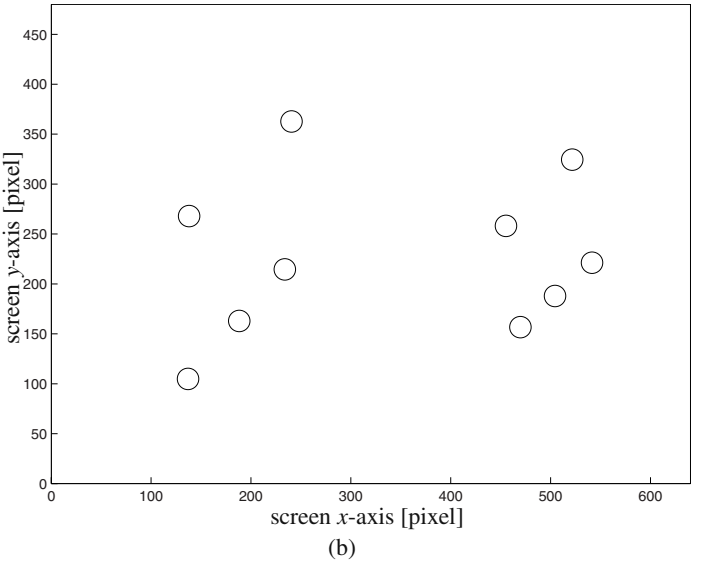


(b)

Fig. 9.1 Example 1: (a) 3D scene with the camera observing three dice; and (b) corresponding camera view (the used object points are the centers of the 10 large dots of the dice)



(a)



(b)

Fig. 9.2 Example 2: (a) 3D scene with the camera observing three dice; and (b) corresponding camera view (the used object points are the centers of the 10 large dots of the dice)

Table 9.2 Example 2: upper and lower bounds of the worst-case location errors for the object points in Figure 9.2

δ [pixels]	$s_r^-(\delta)$ [deg]	$s_r^+(\delta)$ [deg]	$s_t^-(\delta)$ [mm]	$s_t^+(\delta)$ [mm]
0.5	0.141	0.703	0.185	0.789
1.0	0.473	1.405	0.996	1.577
1.5	1.089	2.108	1.492	2.367
2.0	1.895	2.812	1.982	3.158

9.4.2 Example 2

Here we consider the situation shown in Figure 9.2. Again, the chosen object points are the centers of the ten large dots. The screen size and the camera intrinsic parameters are as in Example 1. Table 9.2 shows the upper bounds $s_r^+(\delta)$ and $s_t^+(\delta)$ in (9.24)–(9.25) and the lower bounds $s_r^-(\delta)$ and $s_t^-(\delta)$ in (9.27) for some values of δ.

9.4.3 Example 3

Let us consider the real image shown in Figure 9.3 where twelve object points are observed. The 3D points have been estimated by acquiring a second image from another location, and then performing a standard 3D reconstruction of the object (the camera pose has been estimated through the essential matrix with an estimate

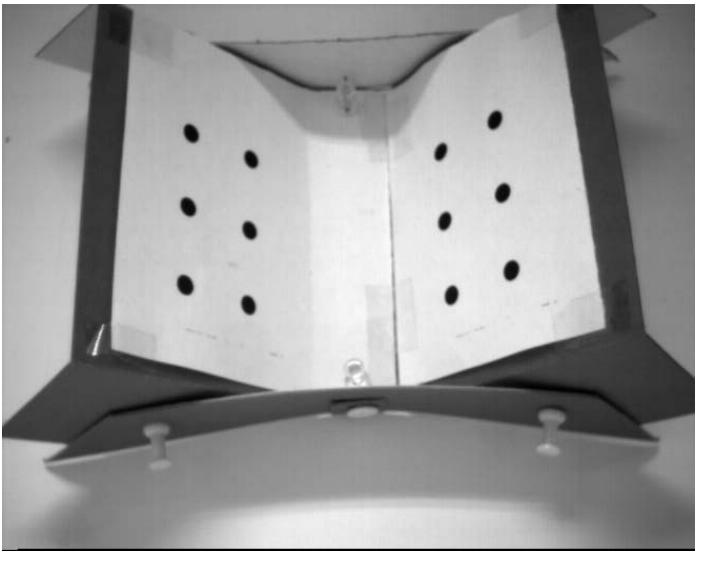


Fig. 9.3 Example 3: real image with twelve observed points

of the camera intrinsic parameters and the knowledge of the distance among two points in the 3D space, in order to estimate the norm of the translation). The screen size is 659×493 pixels, and the estimate of the intrinsic parameters matrix is

$$\mathbf{A} = \begin{pmatrix} 624.2 & 1.035 & 318.1 \\ 0 & 609.7 & 246.3 \\ 0 & 0 & 1 \end{pmatrix}.$$

Table 9.3 shows the upper bounds $s_r^+(\delta)$ and $s_t^+(\delta)$ in (9.24)–(9.25) and the lower bounds $s_r^-(\delta)$ and $s_t^-(\delta)$ in (9.27) for some values of δ.

Table 9.3 Example 3: upper and lower bounds of the worst-case location errors for the object points in Figure 9.3

δ [pixels]	$s_r^-(\delta)$ [deg]	$s_r^+(\delta)$ [deg]	$s_t^-(\delta)$ [mm]	$s_t^+(\delta)$ [mm]
0.5	0.164	1.102	1.091	1.701
1.0	0.329	2.342	2.339	3.399
1.5	0.494	3.496	3.376	5.108
2.0	0.662	4.733	4.593	6.828

9.5 Conclusion

This chapter has addressed the estimation of the worst-case robot positioning error introduced by image measurement errors, which is a fundamental problem in visual servoing. Specifically, it has been shown how upper and lower bounds of this worst-case error can be computed by solving optimization problems over polynomials with LMIs and barrier functions. Future work will investigate the possibility of improving the technique presented in this chapter in order to establish and guarantee exactness of the derived bounds.

Acknowledgements. This work was supported in part by the Research Grants Council of Hong Kong Special Administrative Region under Grants HKU711208E and HKU712808E.

References

[1] Allotta, B., Fioravanti, D.: 3D motion planning for image-based visual servoing tasks. In: Proc. IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain (2005)

[2] Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

[3] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman, D., Hager, G., Morse, A. (eds.) The confluence of vision and control, pp. 66–78. Springer, Heidelberg (1998)

[4] Chaumette, F., Hutchinson, S.: Visual servo control, part I: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)

[5] Chaumette, F., Hutchinson, S.: Visual servo control, part II: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)
[6] Chesi, G.: Visual servoing path-planning via homogeneous forms and LMI optimizations. IEEE Trans. on Robotics 25(2), 281–291 (2009)
[7] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Solving quadratic distance problems: an LMI-based approach. IEEE Trans. on Automatic Control 48(2), 200–212 (2003)
[8] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems. Springer, Heidelberg (2009)
[9] Chesi, G., Hashimoto, K., Prattichizzo, D., Vicino, A.: Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Trans. on Robotics 20(5), 908–913 (2004)
[10] Chesi, G., Hung, Y.S.: Global path-planning for constrained and optimal visual servoing. IEEE Trans. on Robotics 23(5), 1050–1060 (2007)
[11] Chesi, G., Hung, Y.S.: Image noise induced errors in camera positioning. IEEE Trans. on Pattern Analysis and Machine Intelligence 29(8), 1476–1480 (2007)
[12] Chesi, G., Prattichizzo, D., Vicino, A.: Straight line path-planning in visual servoing. Journal of Dynamic Systems, Measurement and Control 129(4), 541–543 (2007)
[13] Chesi, G., Tesi, A., Vicino, A., Genesio, R.: On convexification of some minimum distance problems. In: European Control Conf. Karlsruhe, Germany (1999)
[14] Chesi, G., Vicino, A.: Visual servoing for large camera displacements. IEEE Trans. on Robotics 20(4), 724–735 (2004)
[15] Chesi, G., Yung, H.L.: Performance limitation analysis in visual servo systems: bounding the location error introduced by image points matching. In: IEEE Int. Conf. on Robotics and Automation, Kobe, Japan, pp. 695–700 (2009)
[16] Corke, P.I., Hutchinson, S.: A new partitioned approach to image-based visual servo control. IEEE Trans. on Robotics and Automation 17(4), 507–515 (2001)
[17] Cowan, N.J., Chang, D.E.: Geometric visual servoing. IEEE Trans. on Robotics 21(6), 1128–1138 (2005)
[18] Gans, N., Hutchinson, S.: Stable visual servoing through hybrid switched-system control. IEEE Trans. on Robotics 23(3), 530–540 (2007)
[19] Hashimoto, K.: A review on vision-based control of robot manipulators. Advanced Robotics 17(10), 969–991 (2003)
[20] Hashimoto, K., Kimoto, T., Ebine, T., Kimura, H.: Manipulator control with image-based visual servo. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2267–2272 (1991)
[21] Hutchinson, S., Hager, G., Corke, P.: A tutorial on visual servo control. IEEE Trans. on Robotics and Automation 12(5), 651–670 (1996)
[22] Kyrki, V.: Control uncertainty in image-based visual servoing. In: Proc. IEEE Int. Conf. on Robotics and Automation, Kobe, Japan, pp. 1516–1521 (2009)
[23] Kyrki, V., Kragic, D., Christensen, H.: Measurement errors in visual servoing. In: Proc. IEEE Int. Conf. on Robotics and Automation, New Orleans, Louisiana, pp. 1861–1867 (2004)
[24] Malis, E.: Visual servoing invariant to changes in camera-intrinsic parameters. IEEE Trans. on Robotics and Automation 20(1), 72–81 (2004)
[25] Malis, E., Chaumette, F., Boudet, S.: 2 1/2 D visual servoing. IEEE Trans. on Robotics and Automation 15(2), 238–250 (1999)
[26] Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Trans. on Robotics and Automation 18(4), 534–549 (2002)

[27] Park, J., Chung, M.: Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. IEEE Trans. on Robotics and Automation 19(2), 250–258 (2003)
[28] Schramm, F., Morel, G.: Ensuring visibility in calibration-free path planning for image-based visual servoing. IEEE Trans. on Robotics 22(4), 848–854 (2006)
[29] Tahri, O., Chaumette, F.: Point-based and region-based image moments for visual servoing of planar objects. IEEE Trans. on Robotics 21(6), 1116–1127 (2005)
[30] Thuilot, B., Martinet, P., Cordesses, L., Gallice, J.: Position based visual servoing: keeping the object in the field of vision. In: Proc. IEEE Int. Conf. on Robotics and Automation, Washington, D.C, pp. 1624–1629 (2002)

Chapter 10
Multicriteria Analysis of Visual Servos through Rational Systems, Biquadratic Lyapunov Functions, and LMIs

Patrick Danès, Daniel F. Coutinho, and Sylvain Durola

Abstract. This chapter outlines a generic method to the analysis of eye-in-hand position-based or image-based visual servos. This analysis is said to be "multicriteria" as both the convergence and the fulfillment of important constraints can be assessed, including the target visibility, the avoidance of actuators' saturations, and the exclusion of 3D areas. The field of nonlinear "rational" systems is first shown to constitute a sound and versatile framework to the problem. The fundamentals of a solution based on Lyapunov theory are overviewed next, together with the noteworthy difficulties raised by robotics. Constructive results are finally presented, on the basis of biquadratic or piecewise-biquadratic Lyapunov functions, leading to feasibility/optimization programs subject to linear matrix inequalities (LMIs). A case study illustrates the approach.

10.1 Introduction

Visual servoing has been a very active field of research for the last two decades [4]. Indeed, visual servos are generally more robust to modeling and measurement errors than proprioceptive control, and participate to the perspective of enabling robots to evolve in dynamically changing environments. Within the eye-in-hand approaches, two recurrent strategies can be exhibited depending on the data transmitted to the controller. On the one hand, position-based (or 3D) servos define the controlled variable as the relative situation of the target with respect to the camera. As the quality

Patrick Danès and Sylvain Durola
CNRS; LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France; and
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
e-mail: {patrick.danes,sylvain.durola}@laas.fr

Daniel F. Coutinho
Group of Automation and Control Systems, PUC-RS, Av. Ipiranga 6681,
Porto Alegre, RS 90619, Brazil
e-mail: dcoutinho@pucrs.br

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 169–188.
springerlink.com

of the pose estimates required by such schemes has a decisive effect on their stability and accuracy, their robustness to calibration or measurement errors may happen to be poor. In addition, ensuring the visibility of the target at the controller design stage is a nontrivial issue. On the other hand, image-based (or 2D) servos [14] state the control problem in terms of some visual features extracted from the image, *e.g.* points, lines, moments, *etc*, which intrinsically makes them more robust to calibration errors and image noise. Yet, they present two drawbacks, all the more significant as the displacement to perform is large: a satisfactory apparent motion in the image may correspond to a contorted 3D trajectory, and the camera may converge to a "local minimum" distinct from the desired goal.

In view of the above, many approaches have been developed towards the analysis or synthesis of visual servos in regard to criteria embracing convergence, visibility constraints, actuators saturations, singularity avoidance and 3D effective motion. To cite few, stability analyses are proposed in [3, 17, 22] for specific image-based and hybrid strategies. Visibility constraints can be ensured through path-planning [20], navigation functions [10], as well as partitioned or switching strategies [7, 6]. Joint position limits are handled through constrained quadratic optimization in [15]. Within the task function framework, they are considered together with singularities avoidance in [19] and with visibility constraints in [23] thanks to LMI optimization.

The work developed hereafter proposes a generic approach to the multicriteria analysis of a large class of position-based and image-based servos. It is organized as follows. Section 10.2 introduces the nonlinear rational systems framework and argues its suitability to the analysis and synthesis problems. Section 10.3 outlines the foundations of the method, namely elements of the Lyapunov theory and LMIs. After detailing the multicriteria analysis method in Section 10.4, a case study is considered in Section 10.5. A conclusion ends the chapter.

Notation. Scalars, vectors, and matrices are respectively denoted as in x, $\mathbf{x}$, X. The zero vector, the identity and zero matrices are respectively termed $\mathbf{0}$, I and O, and may be subscripted by their dimensions. The transpose operator is represented by $'$. The notation $\mathsf{M} > 0$ (resp. $\mathsf{M} \geq 0$) means that the matrix M is symmetric and positive definite (resp. semidefinite.) For block matrices, the symbol $\star$ stands for symmetric blocks outside the main diagonal. Given a polytope $\mathscr{X}$, $\mathscr{V}(\mathscr{X})$ denotes the set of all its vertices. For any positive integer N, Ξ_N stands for the set $\{1,\ldots,N\}$.

10.2 The Rational Systems Framework to Visual Servos Multicriteria Analysis and Synthesis

Visual servos which aim to drive a perspective camera to a unique relative situation with respect to a still target in a static environment are considered. Dynamic effects in the camera motion are neglected, and dedicated spots serve as visual features. A state space model uniting 3D and 2D schemes is first set up. Therein, the actuators, the sensor and the image processing system are supposed perfect and instantaneous, so they do not appear. Nonlinear rational systems are then shown to constitute a sound and versatile framework to the multicriteria analysis.

10.2.1 State Space Formulation

This section first introduces the notations underlying the mathematical modeling. Then, an open-loop and a closed-loop state space model are drawn up, enabling the statement of the problem.

10.2.1.1 Frames

The forthcoming state space model hinges on three frames. $F_O = (O, \overrightarrow{x_O}, \overrightarrow{y_O}, \overrightarrow{z_O})$ is associated to the world. $F_S = (S, \overrightarrow{x_S}, \overrightarrow{y_S}, \overrightarrow{z_S})$ is rigidly linked to the camera, with S the optical center and $\overrightarrow{z_S}$ on the optical axis. The third frame $F_T = (T, \overrightarrow{x_T}, \overrightarrow{y_T}, \overrightarrow{z_T})$, attached to the target, is defined as the situation to be reached by F_S. The target is fitted with M spots $T_1, T_2, \ldots T_M$, arranged so that to any configuration of their perspective projections $S_1, S_2, \ldots S_M$ onto the camera image plane corresponds a unique sensor-target relative situation. Let $(\overrightarrow{ST_i})_{(F_S)} = (x_i \; y_i \; z_i)'$ and $(\overrightarrow{TT_i})_{(F_T)} = (a_i \; b_i \; c_i)'$ be the coordinates of T_i, $i = 1, \ldots, M$, in frames F_S and F_T. The metric coordinates $(\overrightarrow{SS_i})_{(F_S)} = (\overline{x}_i \; \overline{y}_i \; f)'$ of S_i, $i = 1, \ldots, M$, in frame F_S thus satisfy $\overline{x}_i = f\frac{x_i}{z_i}$ and $\overline{y}_i = f\frac{y_i}{z_i}$, with f the camera focal length. In addition, the reference values of $\overline{x}_i$ and $\overline{y}_i$ write as $\overline{x}_i^* = f\frac{a_i}{c_i}$ and $\overline{y}_i^* = f\frac{b_i}{c_i}$.

10.2.1.2 Open-loop State Space Model

The way the 6 degrees of freedom (DOF) camera is caused to move and its interaction with the environment are described by an open-loop state space model. In the considered kinematic context, its control input is the velocity screw $\mathbf{u} \triangleq (v_x \; v_y \; v_z \; \omega_x \; \omega_y \; \omega_z)'$, with $(v_x \; v_y \; v_z)'$ and $(\omega_x \; \omega_y \; \omega_z)'$ the entries in F_S of the translational and rotational velocities of F_S with respect to F_O.

As every variable of the system in open-loop is a memoryless function of the sensor-target relative situation, the state vector can be set to $\mathbf{x} = (\mathbf{t}' \; \mathbf{r}')'$ with $\mathbf{t}$ (resp. $\mathbf{r}$) a parametrization of the relative translation (resp. relative attitude) between F_S and F_T. The subvectors $\mathbf{t}$ and $\mathbf{r}$ can respectively be made up with the entries $(\overrightarrow{ST})_{(F_S)} = (t_x \; t_y \; t_z)'$ in frame F_S of the vector joining S and T, and with the Bryant angles[1] $(\lambda \; \mu \; \nu)'$ turning $(\overrightarrow{x_S}, \overrightarrow{y_S}, \overrightarrow{z_S})$ into $(\overrightarrow{x_T}, \overrightarrow{y_T}, \overrightarrow{z_T})$. The output vector $\mathbf{y}$ is defined as the input to the controller. For a position-based scheme, $\mathbf{y}$ is equal to $\mathbf{x}$. When considering an image-based servo, one sets $\mathbf{y} = \mathbf{s} - \mathbf{s}^*$, where $\mathbf{s} \triangleq (\overline{x}_1 \; \overline{y}_1 \; \cdots \; \overline{x}_M \; \overline{y}_M)'$ depicts the projection of the visual features, and $\mathbf{s}^*$ is its reference value.

The state equation, which explains the effect of the velocity screw onto the relative sensor-target situation, is obtained from rigid body kinematics. For 2D servos, the output equation accounts for the interaction between the sensor-target relative situation and the coordinates of the features' perspective projections. The state space model, detailed in the companion report of [11], has the general form

[1] $\mathbf{r}$ is single-valued ($\mu \neq \pm\frac{\pi}{2}$, $\overrightarrow{z_S} \not\perp \overrightarrow{z_T}$) as soon as a visibility constraint is added to the problem.

$$\dot{\mathbf{x}} = \mathsf{F}(\mathbf{x})\mathbf{u}; \quad \left| \begin{array}{ll} \mathbf{y} = \mathbf{x} & \text{(3D servo)} \\ \text{or} & \\ \mathbf{y} = \left(\overline{x}_1 - \overline{x}_1^* \;\; \overline{y}_1 - \overline{y}_1^* \;\; \cdots \;\; \overline{x}_M - \overline{x}_M^* \;\; \overline{y}_M - \overline{y}_M^*\right)' = \mathsf{H}(\mathbf{x}). & \text{(2D servo)} \end{array} \right. \tag{10.1}$$

10.2.1.3 Closed-loop Model and Problem Statement

Let a visual feedback with state vector $\mathbf{x_c}$ and no external input be connected to (10.1). Denote by $\tilde{\mathbf{x}} \triangleq (\mathbf{x}' \; \mathbf{x_c}')' \in \mathbb{R}^{\tilde{n}}$ the state vector of the consequent autonomous closed-loop system. The camera converges to the reference situation whenever the closed-loop equilibrium $\tilde{\mathbf{x}}^* = \mathbf{0}$ is asymptotically stable. This state space formulation prevents "local minima" [3], *i.e.* convergence to poses such that $\mathbf{u} = \mathbf{0}$ while $\mathbf{s} \neq \mathbf{s}^*$. Instead, the convergence of $\mathbf{s}$ to $\mathbf{s}^*$ is a consequence of the attraction of $\tilde{\mathbf{x}}^* = \mathbf{0}$.

Moreover, as all the variables depicting the closed-loop system come as memoryless functions of $\tilde{\mathbf{x}}$, the fulfillment of the other criteria can be turned into the boundedness of some functions $\zeta_j(\tilde{\mathbf{x}})$ by suitable intervals $[\underline{\zeta}_j, \overline{\zeta}_j]$. For instance, the visual features' projections can be restricted to the limits of the camera image plane by defining $\zeta_j = s_j - s_j^*$, including for 3D servos. Actuators saturations can be dealt with as well, *e.g.* by defining some ζ_j's as entries of the velocity screw $\mathbf{u}$ or norms of subvectors extracted from $\mathbf{u}$. 3D constraints, such as constraining the camera motion inside a corridor, can be handled even for 2D servos by bounding some distances $\zeta_j = d_{3\mathrm{D}_j}$. Last, imposing bounds on the control signal $\mathbf{u}$ or the differences $s_j - s_j^*$ enable the avoidance of differential singularities in the loop transfers, *e.g.* when using some 2D "inverse Jacobian" control schemes.

Without loss of generality, each so-called *additional variable* $\zeta_j(.)$, $j \in \Xi_J$, is defined so that $\zeta_j(\mathbf{0}) = \mathbf{0}$.

10.2.2 The Rational Systems Framework

In the following, the multicriteria visual servoing problem is recast into the rational systems framework, and the induced potentialities are analyzed.

10.2.2.1 Rewriting the Multicriteria Analysis/Control Problem

Definition 10.1 (Rational System [13]). *A system is said* rational *if it is defined by*

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathsf{A}(\mathbf{x},\chi) & \mathsf{B}(\mathbf{x},\chi) \\ \mathsf{C}(\mathbf{x},\chi) & \mathsf{D}(\mathbf{x},\chi) \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{u} \end{pmatrix}, \tag{10.2}$$

where $\mathbf{x} \in \mathscr{X} \subset \mathbb{R}^{n_\mathbf{x}}$ *is the state vector and* $\chi \in \mathscr{X}_\chi \subset \mathbb{R}^{n_\chi}$ *is the vector of uncertain parameters;* $\mathscr{X}$ *and* $\mathscr{X}_\chi$ *are given polytopic sets containing the origin* $\mathbf{0}$*; and* $\mathsf{A}(.,.)$, $\mathsf{B}(.,.)$, $\mathsf{C}(.,.)$, $\mathsf{D}(.,.)$ *are rational matrix functions of* $(\mathbf{x},\chi)$ *well-defined on* $\mathscr{X}\times\mathscr{X}_\chi$, i.e. *with no singular entries for all* $(\mathbf{x},\chi) \in \mathscr{X}\times\mathscr{X}_\chi$.

The open-loop system (10.1) involves all the attitude coordinates in trigonometric functions. So, it can be easily turned into a rational form, *e.g.* by redefining

$\mathbf{x} \triangleq \left(t_x\ t_y\ t_z\ \tan(\frac{\lambda}{2})\ \tan(\frac{\mu}{2})\ \tan(\frac{\nu}{4})\right)'$. Connecting the obtained model (10.2) (with zero matrix functions $\mathsf{A}(.,.)$ and $\mathsf{D}(.,.)$ therein) with a rational controller

$$\begin{pmatrix} \dot{\mathbf{x_c}} \\ \mathbf{u} \end{pmatrix} = \begin{pmatrix} \mathsf{K_c}(\mathbf{x},\mathbf{x_c})\ \mathsf{K_{cy}}(\mathbf{x},\mathbf{x_c}) \\ \mathsf{K_u}(\mathbf{x},\mathbf{x_c})\ \mathsf{K_{uy}}(\mathbf{x},\mathbf{x_c}) \end{pmatrix} \begin{pmatrix} \mathbf{x_c} \\ \mathbf{y} \end{pmatrix} \tag{10.3}$$

of state vector $\mathbf{x_c} \in \mathscr{X}_c \subset \mathbb{R}^{n_c}$, with $\mathscr{X}_c$ a given polytope enclosing $\mathbf{0}$, leads to the autonomous closed-loop rational system

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathsf{A}}(\tilde{\mathbf{x}},\chi)\tilde{\mathbf{x}},\ \ \tilde{\mathbf{x}} = (\mathbf{x}'\ \mathbf{x_c}')' \in \tilde{\mathscr{X}} \triangleq \mathscr{X}\times\mathscr{X}_c \subset \mathbb{R}^{\tilde{n}},\ \ \tilde{n} = n_{\mathbf{x}} + n_c,\ \ \mathbf{0} \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi. \tag{10.4}$$

$\tilde{\mathsf{A}}(.,.)$ is assumed well-defined on $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$. As outlined above, the asymptotic stability of the equilibrium $\tilde{\mathbf{x}}^* = \mathbf{0}$ of (10.4) whatever the uncertainty χ is sufficient to the convergence of the camera from initial poses next to the goal. This condition is in essence local, and must be complemented by regional stability considerations so as to cope with farther initial sensor-target situations.

The relationships on the additional variables related to the other criteria can be turned into

$$\forall j \in \Xi_J,\ \forall \chi \in \mathscr{X}_\chi,\ \zeta_j = \mathbf{Z}_j{}'(\tilde{\mathbf{x}},\chi)\tilde{\mathbf{x}} \in [\underline{\zeta}_j, \overline{\zeta}_j], \tag{10.5}$$

with each $\mathbf{Z}_j(.,.)$ a rational column vector function well-defined on $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$.

10.2.2.2 Potentialities for Visual Servoing

The general equation (10.3) depicts a dynamic gain-scheduled nonlinear controller, whose parameters are set on-the-fly to rational functions of the relative sensor-target situation parametrization $\mathbf{x}$ and/or the controller state vector $\mathbf{x_c}$. Noticeably, (10.3) can specialize into simpler visual feedbacks, such as dynamic linear 3D or 2D schemes, and encompasses most "classical" strategies. For instance, the classical "inverse 3D-Jacobian" schemes $u = -\lambda \mathsf{B}^{-1}(0)\mathbf{x}$ and $u = -\lambda \mathsf{B}^{-1}(\mathbf{x})\mathbf{x}$, with $\mathsf{B}(.)$ such that $\dot{\mathbf{x}} = \mathsf{B}(\mathbf{x})\mathbf{u}$, respectively correspond to a linear static state feedback $u = \mathsf{K}\mathbf{x}$ and to a nonlinear rational static state feedback $u = \mathsf{K}(\mathbf{x})\mathbf{x}$. Similarly, the "inverse 2D-Jacobian" controllers $\mathbf{u} = -\lambda[\mathsf{J}(\mathbf{s}^*,\mathbf{z}^*)]^+(\mathbf{s}-\mathbf{s}^*)$ and $\mathbf{u} = -\lambda[\mathsf{J}(\mathbf{s},\mathbf{z})]^+(\mathbf{s}-\mathbf{s}^*)$, with $\mathsf{J}(.,.)$ the interaction matrix defined from $\dot{\mathbf{s}} = \mathsf{J}(\mathbf{s},\mathbf{z})\mathbf{u}$, can also be dealt with. Indeed, $\mathsf{J}(.,.)$ is a rational function of $\mathbf{s}$ and of the vector $\mathbf{z}$ made up with the depths $z_i = \overrightarrow{ST_i}.\overrightarrow{z_S}$, so that these 2D servos respectively correspond to a linear static output feedback $\mathbf{u} = \mathsf{K}\mathbf{y}$ and to a nonlinear rational static output feedback $\mathbf{u} = \mathsf{K}(\mathbf{x})\mathbf{y}$.

Last, the problem statement can be enriched. Dynamic effects can be taken into account in the open-loop model by building $\mathbf{u}$ with the forces and torques which cause the camera motion, and by augmenting $\mathbf{x}$ with velocities. A finer modeling of the camera can be handled as well. Uncertainties affecting rationally the camera parameters, the measurements, or the target model coefficients a_i, b_i, c_i, can be also inserted. The suggested framework can constitute a sound basis to the definition of a "standard problem" [25] of visual servoing, through the introduction of a penalized output z (playing in some way the role of a task function [21]) and of a criterion on z or on the transfer from a relevant input signal w to z.

10.3 Theoretical Foundations of the Method

Lyapunov theory and LMIs constitute the cornerstone of the proposed stability analysis of the rational closed-loop servo (10.4) subject to the rational constraints (10.5). The fundamentals issues are hereafter outlined.

10.3.1 Elements of Lyapunov Theory

This section introduces the concept of multicriteria basin of attraction, and discusses the difficulties raised by robotics in its calculation.

10.3.1.1 Fundamentals of the Definition of a Lyapunov Function

Definition 10.2 (Multicriteria Basin of Attraction). *A set $\tilde{\mathscr{E}}$ is said a* multicriteria basin of attraction *if it is a domain of initial conditions from which the state trajectories of the closed-loop system* (10.4) *converge to the equilibrium $\tilde{\mathbf{x}}^* = \mathbf{0}$ while wholly lying in the admissible subset of the state space defined by the criteria* (10.5)*.*

A function $V(.,.) : \tilde{\mathscr{X}} \times \mathscr{X}_\chi \longrightarrow \mathbb{R}$ is sought for, such that $\tilde{\mathscr{E}} \triangleq \{\tilde{\mathbf{x}} \ : \ V(\tilde{\mathbf{x}},\chi) \leq 1, \ \forall \chi \in \mathscr{X}_\chi\}$ is a multicriteria basin of attraction. The type of $V(.,.)$ is selected beforehand. Its DOF constitute a matrix P, to be tuned so that $V(.,.) = V_\mathsf{P}(.,.)$ satisfies the following three rules:

1. $V_\mathsf{P}(.,.)$ is a Lyapunov function on $\tilde{\mathscr{X}} \times \mathscr{X}_\chi$ so that $\tilde{\mathbf{x}}^* = \mathbf{0}$ is locally asymptotically stable, *i.e.* $V_\mathsf{P}(.,.)$ is continuously differentiable and

$$\forall (\tilde{\mathbf{x}},\chi) \in \left(\tilde{\mathscr{X}} \backslash \{\mathbf{0}\}\right) \times \mathscr{X}_\chi, V_\mathsf{P}(\tilde{\mathbf{x}},\chi) > 0; \qquad \forall \chi \in \mathscr{X}_\chi, V_\mathsf{P}(\mathbf{0},\chi) = 0; \quad (10.6)$$

$$\forall (\tilde{\mathbf{x}},\chi) \in \left(\tilde{\mathscr{X}} \backslash \{\mathbf{0}\}\right) \times \mathscr{X}_\chi, \dot{V}_\mathsf{P}(\tilde{\mathbf{x}},\chi) < 0; \qquad \forall \chi \in \mathscr{X}_\chi, \dot{V}_\mathsf{P}(\mathbf{0},\chi) = 0; \quad (10.7)$$

2. $\tilde{\mathscr{E}}$ is a basin of attraction of $\tilde{\mathbf{x}}^* = 0$ for the unconstrained problem as soon as

$$\forall (\tilde{\mathbf{x}},\chi) : V_\mathsf{P}(\tilde{\mathbf{x}},\chi) \leq 1, \ \tilde{\mathbf{x}} \in \tilde{\mathscr{X}}; \quad (10.8)$$

3. the boundedness (10.5) of the additional variables is ensured when

$$\forall j \in \Xi_J, \ \forall (\tilde{\mathbf{x}},\chi) : V_\mathsf{P}(\tilde{\mathbf{x}},\chi) \leq 1, \ \underline{\zeta}_j \leq \mathbf{Z}_j{}'(\tilde{\mathbf{x}},\chi)\tilde{\mathbf{x}} \leq \overline{\zeta}_j. \quad (10.9)$$

Figure 10.1 illustrates the above conditions when $V(.)$ and $\mathbf{Z}_j(.)$ depend only on $\tilde{\mathbf{x}}$. Therein, $\tilde{\mathscr{A}}$ terms the admissible set and is defined by $\tilde{\mathscr{A}} \triangleq \{\tilde{\mathbf{x}} \ : \ \mathbf{Z}_j{}'(\tilde{\mathbf{x}})\tilde{\mathbf{x}} \in [\underline{\zeta}_j, \overline{\zeta}_j]\}$. Some level sets of the Lyapunov function are sketched, together with the basin $\tilde{\mathscr{E}} \triangleq \{\tilde{\mathbf{x}} \ : \ V(\tilde{\mathbf{x}}) \leq 1\}$. Note that $\tilde{\mathscr{E}}$ lies both into $\tilde{\mathscr{X}}$ and $\tilde{\mathscr{A}}$.

10.3.1.2 Additional Guidelines

Additional arguments can lead to more purposeful multicriteria basins of attraction. For instance, when the aim is to analyze the feedback system for initial

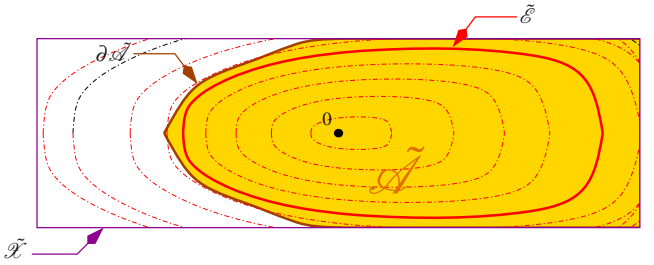


Fig. 10.1 Illustration of the basics of a Lyapunov solution

sensor-target situations in a given set $\mathscr{X}_0$, extra conditions must be put on the matrix P in addition to (10.6)–(10.9) so as to ensure that $\tilde{\mathscr{X}}_0 \triangleq \mathscr{X}_0 \times \mathscr{X}_{c0} \in \tilde{\mathscr{E}}$, with $\mathscr{X}_{c0}$ the domain of checked initial values of the controller state vector. An optimization problem on P under the above constraints an also guide the search. For instance, $\min_{\mathsf{P}} f(\mathsf{P})$ enables the heuristic maximization of the "size" of $\tilde{\mathscr{E}}$ defined from $V(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}'\mathsf{P}\tilde{\mathbf{x}}$, $\mathsf{P} > 0$, if $f(\mathsf{P})$ is set to trace(P) [2]. In the same vein, considering $\min_{\mathsf{P}} \max_{\chi} \max_h V(\lambda_h, \chi)$ can help in heuristically maximizing the extent of $\tilde{\mathscr{E}}$ towards a set of selected points $\{\lambda_h\} \subset \tilde{\mathscr{X}}$ [12].

10.3.1.3 Difficulties Raised by the Robotics Context

Despite the above control problem looks fairly usual, the robotics context raises sharp issues. Indeed, the admissible areas of the state space in regard to the criteria are seldom symmetric with respect to $\mathbf{0}$, nor convex, nor even connected. A fundamental consequence is that ensuring (10.9) may be far too conservative for some classes of Lyapunov functions. For instance, it was proved in [11] that using quadratic Lyapunov functions of the form $V(\tilde{\mathbf{x}}) = \tilde{\mathbf{x}}'\mathsf{P}\tilde{\mathbf{x}}$, with $\mathsf{P} > 0$, leads to extremely poor conclusions even for very simple problems, unless nontrivial theoretical enhancements are devised. Widening the class of Lyapunov function candidates can be a solution, yet there is generally a balance between the obtained versatility and the induced computational complexity. The easiest that can be done given a class of Lyapunov functions is to define $\tilde{\mathscr{E}}$ as the union $\tilde{\mathscr{E}} \triangleq \cup_{r \in \Xi_R} (\tilde{\mathscr{E}})_r$ of R multicriteria basins of attraction $(\tilde{\mathscr{E}})_r$ independently computed under the assumption of polytopes $(\tilde{\mathscr{X}} \times \mathscr{X}_\chi)_r$ and/or of the sets of directions of extent $(\{\lambda_h\})_r$.

Piecewise Lyapunov functions can be envisaged as an alternative with limited complexity. First, $\mathbb{R}^{\tilde{n}}$ is partitioned into S distinct convex cones $\tilde{\mathscr{C}}_1, \ldots, \tilde{\mathscr{C}}_S$ joining at $\mathbf{0}$. Roughly speaking, a set of matrices $\mathsf{P}_1, \ldots, \mathsf{P}_S$ is then sought for so that (i) $V_{\mathsf{P}_s}(.,.)$ is a Lyapunov function on each $\tilde{\mathscr{X}}_s \triangleq \tilde{\mathscr{X}} \cap \tilde{\mathscr{C}}_s$, $s \in \Xi_S$, (ii) the consequent set $\tilde{\mathscr{E}}_s \triangleq \{\tilde{\mathbf{x}} \in \tilde{\mathscr{C}}_s : V(\tilde{\mathbf{x}}, \chi) \leq 1, \forall \chi \in \mathscr{X}_\chi\}$ lies into $\tilde{\mathscr{X}}$ and is admissible in regard to the criteria. Let $V(.,.) : \tilde{\mathscr{X}} \times \mathscr{X}_\chi \longrightarrow \mathbb{R}$ be the piecewise function whose restriction to each $\tilde{\mathscr{X}}_s$ is equal to $V_s(.,.)$, $s \in \Xi_S$. Then $\tilde{\mathscr{E}} = \cup_{s \in \Xi_S} \tilde{\mathscr{E}}_s$ becomes a multicriteria basin of attraction if extra constraints are put on $\mathsf{P}_1, \ldots, \mathsf{P}_S$ so that $V(.,.)$ is continuous on all the interfaces $\partial\tilde{\mathscr{X}}_{s_a s_b}$ between pairs of adjacent cells $\tilde{\mathscr{X}}_{s_a}$ and

$\tilde{\mathscr{X}}_{s_b}$, $(s_a, s_b) \in \Xi_S \times \Xi_S$. To ensure that the time-derivative $\dot{V}(.,.)$ along the system trajectories is well-defined on all $\partial\tilde{\mathscr{X}}_{s_a s_b}$'s, the continuity of $\dot{V}(.,.)$ can be imposed.

10.3.2 Matrix Inequalities and Related Important Lemmas

Definition 10.3 (LMIs [2]). *A constraint* $\mathsf{L}(\mathbf{x})$ *on the real-valued vector or matrix decision variable* $\mathbf{x}$ *is an* LMI *on* $\mathbf{x}$ *if it writes as the negative or positive definiteness of an affine matrix combination of the entries* $x_1, \ldots, x_n$ *of* $\mathbf{x}$*, i.e. if (with* $\mathsf{A}_i = \mathsf{A}_i'$ *given real matrices)*

$$\mathsf{L}(\mathbf{x}) \ : \ \mathsf{A}_0 + x_1\mathsf{A}_1 + \cdots + x_n\mathsf{A}_n \leq 0. \tag{10.10}$$

LMI constraints are convex. Consequently, the feasibility of a set of LMIs as well as the minimum of a convex criterion subject to LMIs are convex problems, whose solutions can be worked out numerically in polynomial time with an arbitrary precision. So, such problems are considered as solved. The versatility of semidefinite programming in engineering was acknowledged more than a decade ago [2, 16].

Thanks to the equivalent representations of (10.4)–(10.5) to be introduced in Section 10.4.1, the rules (10.6)–(10.9) will be turned into inclusion relationships between sets defined by quadratic functions. The following lemmas bridge the gap with LMIs.

Lemma 10.1 (S-procedure [2]). *Consider quadratic functions* $f_l(\xi) \triangleq \begin{pmatrix}\xi\\1\end{pmatrix}' \mathsf{F}_l \begin{pmatrix}\xi\\1\end{pmatrix}$, $l \in \Xi_L$*, with* $\mathsf{F}_l = \mathsf{F}_l'$*. Then*

$$\{\xi \ : \ f_l(\xi) \leq 0, \ \forall l \in \Xi_L\} \subset \{\xi \ : \ f_0(\xi) \leq 0\} \tag{10.11}$$

is true if

$$\exists \tau_1 \geq 0, \ldots, \tau_L \geq 0 \ : \ \forall \xi, \ f_0(\xi) - \sum_{l=1}^{L} \tau_l f_l(\xi) \leq 0. \tag{10.12}$$

When the entries of ξ *are independent,* (10.12) *is equivalent to the LMI on* $\tau_1, \ldots, \tau_L$*:*

$$\tau_1 \geq 0, \ldots, \tau_L \geq 0 \text{ and } \mathsf{F}_0 - \sum_{l=1}^{L} \tau_l \mathsf{F}_l \leq 0. \tag{10.13}$$

If the entries of ξ are related, a less conservative sufficient condition can be got.

Lemma 10.2 ([24, 8]). *Consider two vectors* $\tilde{\mathbf{x}}, \chi$ *in given convex polytopes* $\tilde{\mathscr{X}}, \mathscr{X}_\chi$*. Define a matrix function* $\Sigma_{\tau_l,\mathsf{P},\ldots}(.,.) = \Sigma'_{\tau_l,\mathsf{P},\ldots}(.,.) : \tilde{\mathscr{X}} \times \mathscr{X}_\chi \longrightarrow \mathbb{R}^{n_\sigma \times n_\sigma}$*, with affine dependency on its arguments and on the decision variables* $\tau_l, \mathsf{P}, \ldots$ *Consider the following constraint on a prescribed nonlinear vector function* $\sigma(.,.) : \tilde{\mathscr{X}} \times \mathscr{X}_\chi \longrightarrow \mathbb{R}^{n_\sigma}$*:*

$$\forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}} \times \mathscr{X}_\chi, \ \sigma'(\tilde{\mathbf{x}}, \chi) \Sigma_{\tau_l,\mathsf{P},\ldots}(\tilde{\mathbf{x}}, \chi) \sigma(\tilde{\mathbf{x}}, \chi) \leq 0. \tag{10.14}$$

By convexity, if the LMIs $\Sigma_{\tau_l,\mathsf{P},\ldots}(\tilde{\mathbf{x}}, \chi) \leq 0$ *on* $\tau_l, \mathsf{P}, \ldots$ *hold at all the vertices* $(\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}} \times \mathscr{X}_\chi)$*, then they also hold on* $\tilde{\mathscr{X}} \times \mathscr{X}_\chi$ *and* (10.14) *is satisfied. Yet, if an affine*

matrix function $\Lambda_\sigma(.,.) \in \mathbb{R}^{m_\sigma \times n_\sigma}$ *can be exhibited such that* $\Lambda_\sigma(\tilde{\mathbf{x}},\chi)\sigma(\tilde{\mathbf{x}},\chi) = \mathbf{0}$ *is true on* $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$*, then the following is a less conservative sufficient condition to* (10.14)*:*

$$\exists \mathsf{L} \in \mathbb{R}^{n_\sigma \times m_\sigma} \text{ such that } \forall (\tilde{\mathbf{x}},\chi) \in \mathscr{V}(\tilde{\mathscr{X}}\times\mathscr{X}_\chi),\ \overline{\Sigma}_{\tau_l,\mathsf{P},\ldots,\mathsf{L}}(\tilde{\mathbf{x}},\chi) \leq 0, \qquad (10.15)$$
$$\text{with } \overline{\Sigma}_{\tau_l,\mathsf{P},\ldots,\mathsf{L}}(\tilde{\mathbf{x}},\chi) \triangleq \Sigma_{\tau_l,\mathsf{P},\ldots}(\tilde{\mathbf{x}},\chi) + \mathsf{L}\Lambda_\sigma(\tilde{\mathbf{x}},\chi) + {\Lambda_\sigma}'(\tilde{\mathbf{x}},\chi)\mathsf{L}'.$$

Each constraint $\overline{\Sigma}_{\tau_l,\mathsf{P},\ldots,\mathsf{L}}(\tilde{\mathbf{x}},\chi) \leq 0$ is an LMI on $\tau_l,\mathsf{P},\ldots,\mathsf{L}$ computed at a vertex $(\tilde{\mathbf{x}},\chi)$ of $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$. The matrix function $\Lambda_\sigma(.,.)$ is said a *linear annihilator* of $\sigma(.,.)$.

10.4 Multicriteria Analysis through Biquadratic Lyapunov Functions

Constructive results to the multicriteria analysis of visual servos by biquadratic and piecewise-biquadratic Lyapunov functions are hereafter detailed.

10.4.1 Mathematical Background

Equivalent representations of the closed-loop system and of the constraints are first introduced. Then, the classes of Lyapunov functions candidates are presented.

Definition 10.4 (Differential Algebraic Representations [24, 8]). *A* differential algebraic representation (DAR) *of an uncertain nonlinear rational system is defined by*

$$\begin{cases} \dot{\tilde{\mathbf{x}}} = \mathsf{A}_1\tilde{\mathbf{x}} + \mathsf{A}_2\pi \\ \mathbf{0} = \Omega_1(\tilde{\mathbf{x}},\chi)\tilde{\mathbf{x}} + \Omega_2(\tilde{\mathbf{x}},\chi)\pi, \end{cases} \qquad (10.16)$$

where $\pi = \pi(\tilde{\mathbf{x}},\chi) \in \mathbb{R}^{n_\pi}$ *is a nonlinear vector function of* $(\tilde{\mathbf{x}},\chi)$*,* $\mathsf{A}_1,\mathsf{A}_2$ *are constant matrices and* $\Omega_1(.,.),\Omega_2(.,.)$ *are affine matrix functions. The above representation is well-posed if* $\Omega_2(.,.)$ *is column full rank for all* $(\tilde{\mathbf{x}},\chi) \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi$*. The rational closed loop system* (10.4) *can be turned into* (10.16) *by gathering nonlinear terms in* π *and augmenting this vector so that its entries can be united to* $(\tilde{\mathbf{x}},\chi)$ *through some affine matrix functions* $\Omega_1(.,.)$ *and* $\Omega_2(.,.)$*. Importantly, a DAR is not unique.*

Similarly, the additional variables $\zeta_j = \mathbf{Z}_j{}'(\tilde{\mathbf{x}},\chi)\tilde{\mathbf{x}}$ *defined in* (10.5) *write as [12]*

$$\begin{cases} \zeta_j = \mathbf{K_{1j}}'\tilde{\mathbf{x}} + \mathbf{K_{2j}}'\rho \\ \mathbf{0} = \Upsilon_{1\mathrm{j}}(\tilde{\mathbf{x}},\chi)\tilde{\mathbf{x}} + \Upsilon_{2\mathrm{j}}(\tilde{\mathbf{x}},\chi)\rho, \end{cases} \qquad (10.17)$$

with $\rho = \rho(\tilde{\mathbf{x}},\chi)$ *a nonlinear vector function of* $(\tilde{\mathbf{x}},\chi)$*,* $\mathbf{K_{1j}},\mathbf{K_{2j}}$ *constant vectors,* $\Upsilon_{1\mathrm{j}}(.,.),\Upsilon_{2\mathrm{j}}(.,.)$ *affine matrix functions, and* $\Upsilon_{2\mathrm{j}}(.,.)$ *column full rank on* $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$*.*

The considered classes of Lyapunov function candidates are as follows.

Definition 10.5 (Biquadratic Lyapunov Functions (BQLFs)). BQLFs *candidates for the analysis of* (10.4) *subject to* (10.5) *are defined by*

$$\forall(\tilde{\mathbf{x}},\chi)\in\tilde{\mathscr{X}}\times\mathscr{X}_\chi,\ V(\tilde{\mathbf{x}},\chi)=\phi_1'(\tilde{\mathbf{x}},\chi)\mathsf{P}\phi_1(\tilde{\mathbf{x}},\chi),\ \ \phi_1(\tilde{\mathbf{x}},\chi)\triangleq\left(\Theta(\tilde{\mathbf{x}},\chi)'\ \mathsf{I}_{\tilde{n}}\right)'\tilde{\mathbf{x}},\tag{10.18}$$

where $\Theta(.,.)\in\mathbb{R}^{n_\Theta\times\tilde{n}}$ *terms a linear matrix function selected beforehand, and* $\mathsf{P}=\mathsf{P}'\in\mathbb{R}^{(n_\Theta+\tilde{n})\times(n_\Theta+\tilde{n})}$ *is a constant matrix to be tuned. Importantly, BQLFs can lead to asymmetric nonconvex basins* $\tilde{\mathscr{E}}\triangleq\{\tilde{\mathbf{x}}\ :\ V(\tilde{\mathbf{x}},\chi)\leq 1,\ \forall\chi\in\mathscr{X}_\chi\}$.

Definition 10.6 (Piecewise-biquadratic Lyapunov Functions (PW-BQLFs) [9]). *Partition* $\tilde{\mathscr{X}}\subset\mathbb{R}^{\tilde{n}}$ *into S convex polytopic regions* $\tilde{\mathscr{X}}_1,\ldots,\tilde{\mathscr{X}}_S$ *enclosing* $\mathbf{0}$*, so that* $\tilde{\mathscr{X}}=\cup_{s\in\Xi_S}\tilde{\mathscr{X}}_s$*. Assume that the boundary* $\partial\tilde{\mathscr{X}}_{s_as_b}$ *of two adjacent cells* $\tilde{\mathscr{X}}_{s_a}$ *and* $\tilde{\mathscr{X}}_{s_b}$*,* $s_a\neq s_b\in\Xi_S$*, is a* $(\tilde{n}-1)$*-dimensional polytope. Denote by* Π *the set of pairs of indexes of adjacent cells arranged in increasing order, i.e.*

$$\Pi\triangleq\left\{(s_a,s_b)\in\Xi_S\times\Xi_S\ :\ s_a<s_b\ \textit{and}\ \tilde{\mathscr{X}}_{s_a}\cap\tilde{\mathscr{X}}_{s_b}\neq\{\mathbf{0}\}\right\}.\tag{10.19}$$

To each cell $\tilde{\mathscr{X}}_s$*, associate a BQLF*

$$\forall(\tilde{\mathbf{x}},\chi)\in\tilde{\mathscr{X}}_s\times\mathscr{X}_\chi,\ V_s(\tilde{\mathbf{x}},\chi)=\phi_1'(\tilde{\mathbf{x}},\chi)\mathsf{P}_s\phi_1(\tilde{\mathbf{x}},\chi),\ \ \phi_1(\tilde{\mathbf{x}},\chi)\triangleq\left(\Theta(\tilde{\mathbf{x}},\chi)'\ \mathsf{I}_{\tilde{n}}\right)'\tilde{\mathbf{x}},\tag{10.20}$$

$s\in\Xi_S$*, with* $\mathsf{P}_s=\mathsf{P}_s'\in\mathbb{R}^{(n_\Theta+\tilde{n})\times(n_\Theta+\tilde{n})}$ *a matrix to be determined, and* $\Theta(.,.)\in\mathbb{R}^{n_\Theta\times\tilde{n}}$ *a predefined linear matrix function. Let* $V(.,.):\tilde{\mathscr{X}}\times\mathscr{X}_\chi\longrightarrow\mathbb{R}$ *be the piecewise function whose restriction to each* $\tilde{\mathscr{X}}_s$*,* $s\in\Xi_S$*, is equal to* $V_s(.,.)$*. If, in addition,*

$$\forall(s_a,s_b)\in\Pi,\forall(\tilde{\mathbf{x}},\chi)\in\partial\tilde{\mathscr{X}}_{s_as_b}\times\mathscr{X}_\chi,\ V_{s_a}(\tilde{\mathbf{x}},\chi)=V_{s_b}(\tilde{\mathbf{x}},\chi)\ \textit{and}\ \dot{V}_{s_a}(\tilde{\mathbf{x}},\chi)=\dot{V}_{s_b}(\tilde{\mathbf{x}},\chi),\tag{10.21}$$

then $V(.,.)$ *is said a* PW-BQLF *on* $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$*. Further, a basin of attraction* $\tilde{\mathscr{E}}$ *is defined as the union*

$$\tilde{\mathscr{E}}\triangleq\cup_{s\in\Xi_S}\tilde{\mathscr{E}}_s,\ \textit{with}\ \forall s\in\Xi_S,\ \tilde{\mathscr{E}}_s\triangleq\{\tilde{\mathbf{x}}\in\tilde{\mathscr{X}}_s\ :\ V_s(\tilde{\mathbf{x}},\chi)\leq 1,\forall\chi\in\mathscr{X}_\chi\}.\tag{10.22}$$

10.4.2 LMI Conditions for Multicriteria Analysis Based on BQLFs

To simplify, the uncertainty vector χ is assumed constant[2]. The vector function $\phi_1(\tilde{\mathbf{x}},\chi)=\left(\Theta(\tilde{\mathbf{x}},\chi)'\ \mathsf{I}_{\tilde{n}}\right)'\tilde{\mathbf{x}}$ being introduced in (10.18), define $\Gamma_1=\left(\mathsf{O}_{\tilde{n}\times\tilde{n}_\Theta}\ \mathsf{I}_{\tilde{n}}\right)$ so that $\Gamma_1\phi_1(\tilde{\mathbf{x}},\chi)=\tilde{\mathbf{x}}$, and $\tilde{\Theta}(.,.):\tilde{\mathscr{X}}\times\mathscr{X}_\chi\rightarrow\mathbb{R}^{(n_\Theta+\tilde{n})\times\tilde{n}}$ so that $\dot{\phi}_1(\tilde{\mathbf{x}},\chi)=\tilde{\Theta}(\tilde{\mathbf{x}},\chi)\dot{\tilde{\mathbf{x}}}$. Let the vectors $\mathbf{a}_k$, $k\in\Xi_K$, define the edges of $\tilde{\mathscr{X}}$ by $\tilde{\mathscr{X}}=\{\tilde{\mathbf{x}}:\mathbf{a}_k'\tilde{\mathbf{x}}\leq 1,\ k\in\Xi_K\}$. Some preliminaries are needed to establish LMI sufficient conditions for $\tilde{\mathscr{E}}$ defined within Definition 10.5 to be a multicriteria basin of attraction for the visual servo (10.4) subject to the constraints (10.5).

[2] The method can be easily extended in order to handle time-varying smooth parametric uncertainties in χ, once polytopes enclosing χ and $\dot{\chi}$ are given.

10.4.2.1 Preliminary Results

Local asymptotic stability of $\tilde{\mathbf{x}}^* = \mathbf{0}$. Equations 10.6 and 10.7, which express the positive definiteness of the Lyapunov function $V_{\mathsf{P}}(\tilde{\mathbf{x}},\chi)$ on $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$ and the negative definiteness on $\tilde{\mathscr{X}}\times\mathscr{X}_\chi$ of its derivative along the closed-loop system trajectories, can be straightly recast into constraints like (10.14). Indeed, by defining $\phi_2(\tilde{\mathbf{x}},\chi) \triangleq \left(\phi_1(\tilde{\mathbf{x}},\chi)' \ \pi'\right)'$, one gets

$$(10.6) \Leftrightarrow \forall(\tilde{\mathbf{x}},\chi) \in \left(\tilde{\mathscr{X}}\backslash\{\mathbf{0}\}\right)\times\mathscr{X}_\chi,\ \phi_1'(\tilde{\mathbf{x}},\chi)\mathsf{P}\phi_1(\tilde{\mathbf{x}},\chi) > 0, \tag{10.23}$$

$$(10.7) \Leftrightarrow \forall(\tilde{\mathbf{x}},\chi) \in \left(\tilde{\mathscr{X}}\backslash\{\mathbf{0}\}\right)\times\mathscr{X}_\chi,\ \phi_2'(\tilde{\mathbf{x}},\chi)\mathscr{M}_{(\tilde{\mathbf{x}},\chi)}(\mathsf{P})\phi_2'(\tilde{\mathbf{x}},\chi) < 0, \tag{10.24}$$

with $\mathscr{M}_{(\tilde{\mathbf{x}},\chi)}(\mathsf{P}) \triangleq \begin{pmatrix} \Gamma_1{}'\mathsf{A}_1{}'\tilde{\Theta}'(\tilde{\mathbf{x}},\chi)\mathsf{P}+\mathsf{P}\tilde{\Theta}(\tilde{\mathbf{x}},\chi)\mathsf{A}_1\Gamma_1 & \star \\ \mathsf{A}_2{}'\tilde{\Theta}'(\tilde{\mathbf{x}},\chi)\mathsf{P} & \mathsf{O} \end{pmatrix}$. Notice that (10.24) has a very simple form thanks to the aforementioned definition of the matrix function $\tilde{\Theta}(.,.)$ and of the expression of $\dot{\tilde{\mathbf{x}}}$ in (10.16).

Conditions for $\tilde{\mathscr{E}}$ *to be a basin of attraction of* $\tilde{\mathbf{x}}^* = \mathbf{0}$ *for the unconstrained problem*. Define $\phi_3(\tilde{\mathbf{x}},\chi) \triangleq \left(1 \ \phi_1(\tilde{\mathbf{x}},\chi)'\right)'$. The inclusion relation (10.8) holds if and only if

$$\begin{aligned} &\forall k \in \Xi_K,\ \forall(\tilde{\mathbf{x}},\chi) \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi, \\ &\quad \left(\phi_3'(\tilde{\mathbf{x}},\chi)\begin{pmatrix} -1 & \star \\ \mathbf{0} & \mathsf{P} \end{pmatrix}\phi_3(\tilde{\mathbf{x}},\chi) \le 0\right) \Rightarrow \left(\phi_3'(\tilde{\mathbf{x}},\chi)\begin{pmatrix} -2 & \star \\ \Gamma_1{}'\mathbf{a}_k & \mathsf{O} \end{pmatrix}\phi_3(\tilde{\mathbf{x}},\chi) \le 0\right). \end{aligned} \tag{10.25}$$

By applying the S-procedure (Lemma 10.1), a sufficient condition to (10.25) is

$$\forall k \in \Xi_K,\ \exists\tilde{\eta}_k \ge 0 : \forall(\tilde{\mathbf{x}},\chi) \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi,\ \phi_3'(\tilde{\mathbf{x}},\chi)\begin{pmatrix} -2+\tilde{\eta}_k & \star \\ \Gamma_1{}'\mathbf{a}_k & -\tilde{\eta}_k\mathsf{P} \end{pmatrix}\phi_3(\tilde{\mathbf{x}},\chi) \le 0. \tag{10.26}$$

For each k, the multiplier $\tilde{\eta}_k$ can be forced to be nonzero. So, after the change of variable $\eta_k \triangleq \frac{1}{\tilde{\eta}_k}$, (10.26) can be turned into

$$\forall k \in \Xi_K,\ \exists\eta_k > 0 : \forall(\tilde{\mathbf{x}},\chi) \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi,\ \phi_3'(\tilde{\mathbf{x}},\chi)\begin{pmatrix} -2\eta_k+1 & \star \\ \eta_k\Gamma_1{}'\mathbf{a}_k & -\mathsf{P} \end{pmatrix}\phi_3(\tilde{\mathbf{x}},\chi) \le 0. \tag{10.27}$$

Conditions for $\tilde{\mathscr{E}}$ *to be a multicriteria basin of attraction of* $\tilde{\mathbf{x}}^* = \mathbf{0}$. Further, the set $\tilde{\mathscr{E}}$ is a multicriteria basin of attraction if it also lies in the admissible domain $\tilde{\mathscr{A}} \triangleq \{\tilde{\mathbf{x}} : (10.5) \text{ holds}\}$. This is ensured by (10.9), or, equivalently, by

$$\begin{aligned} &\forall j \in \Xi_J,\ \forall(\tilde{\mathbf{x}},\chi) \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi, \\ &\quad \left(\phi_4'(\tilde{\mathbf{x}},\chi)\mathscr{N}_0(\mathsf{P})\phi_4(\tilde{\mathbf{x}},\chi) \le 0\right) \Rightarrow \begin{cases} \phi_4'(\tilde{\mathbf{x}},\chi)\mathscr{N}_{1,j}\phi_4(\tilde{\mathbf{x}},\chi) \le 0, \\ \phi_4'(\tilde{\mathbf{x}},\chi)\mathscr{N}_{2,j}\phi_4(\tilde{\mathbf{x}},\chi) \le 0, \end{cases} \end{aligned} \tag{10.28}$$

with $\mathscr{N}_0(\mathsf{P}) \triangleq \begin{pmatrix} -1 & \star & \star \\ \mathbf{0} & \mathsf{P} & \star \\ \mathbf{0} & \mathsf{O} & \mathsf{O} \end{pmatrix}$, $\mathscr{N}_{1,j} \triangleq \begin{pmatrix} -2\overline{\zeta}_j & \star & \star \\ \Gamma_1{}'\mathbf{K_{1j}} & \mathsf{O} & \mathsf{O} \\ \mathbf{K_{2j}} & \mathsf{O} & \mathsf{O} \end{pmatrix}$, $\mathscr{N}_{2,j} \triangleq \begin{pmatrix} 2\underline{\zeta}_j & \star & \star \\ -\Gamma_1{}'\mathbf{K_{1j}} & \mathsf{O} & \mathsf{O} \\ -\mathbf{K_{2j}} & \mathsf{O} & \mathsf{O} \end{pmatrix}$ and $\phi_4(\tilde{\mathbf{x}},\chi) \triangleq \left(1 \ \phi_1(\tilde{\mathbf{x}},\chi)' \ \rho(\tilde{\mathbf{x}},\chi)'\right)'$. By the S-procedure (Lemma 10.1), a sufficient condition to (10.28) comes as

$$\forall j \in \Xi_J,\ \exists \tilde{\sigma}_j \geq 0,\ \exists \tilde{\tau}_j \geq 0 : \forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}} \times \mathscr{X}_\chi,$$
$$\begin{cases} \phi_4'(\tilde{\mathbf{x}}, \chi)\big(\mathscr{N}_{1,j} - \tilde{\sigma}_j \mathscr{N}_0(\mathsf{P})\big)\phi_4(\tilde{\mathbf{x}}, \chi) \leq 0, \\ \phi_4'(\tilde{\mathbf{x}}, \chi)\big(\mathscr{N}_{2,j} - \tilde{\tau}_j \mathscr{N}_0(\mathsf{P})\big)\phi_4(\tilde{\mathbf{x}}, \chi) \leq 0. \end{cases} \tag{10.29}$$

For each j, the multipliers $\tilde{\sigma}_j$ and $\tilde{\tau}_j$ cannot be zero as $\tilde{\mathscr{X}}$ is not wholly admissible. Setting $\sigma_j \triangleq \frac{1}{\tilde{\sigma}_j}$ and $\tau_j \triangleq \frac{1}{\tilde{\tau}_j}$ leads to the following equivalent formulation of (10.29), which is also a sufficient condition to (10.9):

$$\forall j \in \Xi_J,\ \exists \sigma_j > 0,\ \exists \tau_j > 0 : \forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}} \times \mathscr{X}_\chi,$$
$$\begin{cases} \phi_4'(\tilde{\mathbf{x}}, \chi)\big(\sigma_j \mathscr{N}_{1,j} - \mathscr{N}_0(\mathsf{P})\big)\phi_4(\tilde{\mathbf{x}}, \chi) \leq 0, \\ \phi_4'(\tilde{\mathbf{x}}, \chi)\big(\tau_j \mathscr{N}_{2,j} - \mathscr{N}_0(\mathsf{P})\big)\phi_4(\tilde{\mathbf{x}}, \chi) \leq 0. \end{cases} \tag{10.30}$$

Inclusion of a given ellipsoidal set $\tilde{\mathscr{X}}_0$ into $\tilde{\mathscr{E}}$. Consider the problem of enclosing a given ellipsoid $\tilde{\mathscr{X}}_0 \triangleq \mathscr{X}_0 \times \mathscr{X}_{c0}$ by the multicriteria basin of attraction $\tilde{\mathscr{E}}$, where $\mathscr{X}_0, \mathscr{X}_{c0}$ respectively term some sets of initial sensor-target situations and initial controller state vectors. Let $\tilde{\mathscr{X}}_0$ write as

$$\tilde{\mathscr{X}}_0 \triangleq \{\tilde{\mathbf{x}} : \tilde{\mathbf{x}} = \hat{\tilde{\mathbf{x}}}_0 + \mathsf{E}\mathbf{z},\ |\mathbf{z}| \leq 1,\ \mathbf{z} \in \mathbb{R}^{n_\mathbf{z}}\} \tag{10.31}$$

with $\hat{\tilde{\mathbf{x}}}_0$ its center and E its "shape matrix". Setting $\phi_5(\tilde{\mathbf{x}}, \chi) \triangleq \begin{pmatrix} 1 & \phi_1(\tilde{\mathbf{x}}, \chi)' & \mathbf{z}' \end{pmatrix}'$ leads to the following trivial formulation of the inclusion $\tilde{\mathscr{X}}_0 \subset \tilde{\mathscr{E}}$:

$$\forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}} \times \mathscr{X}_\chi,\ \big(\phi_5'(\tilde{\mathbf{x}}, \chi)\mathscr{O}_0\phi_5(\tilde{\mathbf{x}}, \chi) \leq 0\big) \Rightarrow \big(\phi_5'(\tilde{\mathbf{x}}, \chi)\mathscr{O}_1\phi_5(\tilde{\mathbf{x}}, \chi) \leq 0\big), \tag{10.32}$$

with $\mathscr{O}_0 \triangleq \begin{pmatrix} -1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathsf{O} & \mathsf{O} \\ \mathbf{0} & \mathsf{O} & \mathsf{I} \end{pmatrix}$ and $\mathscr{O}_1 \triangleq \begin{pmatrix} -1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathsf{P} & \mathsf{O} \\ \mathbf{0} & \mathsf{O} & \mathsf{O} \end{pmatrix}$. By the S-procedure (Lemma 10.1), a sufficient condition to (10.32) is

$$\exists \kappa \geq 0 : \forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}} \times \mathscr{X}_\chi,\ \phi_5'(\tilde{\mathbf{x}}, \chi)\big(\mathscr{O}_1 - \kappa \mathscr{O}_0\big)\phi_5(\tilde{\mathbf{x}}, \chi) \leq 0, \tag{10.33}$$

where κ is readily seen to be nonzero. We are now ready for the main theorem.

Theorem 10.1 (Multicriteria Analysis via BQLFs). *$\tilde{\mathscr{E}}$ defined within Definition 10.5 is a multicriteria basin of attraction for the visual servo* (10.4) *subject to the constraints* (10.5) *and encloses the ellipsoid of initial conditions defined by* (10.31) *if the LMIs on the matrices L, W, $\{\mathsf{Y}_k\}_{k\in\Xi_K}$, $\{\mathsf{F}_j\}_{j\in\Xi_J}$, $\{\mathsf{G}_j\}_{j\in\Xi_J}$, K, on the positive scalars $\{\eta_k\}_{k\in\Xi_K}$, $\{\sigma_j\}_{j\in\Xi_J}$, $\{\tau_j\}_{j\in\Xi_J}$, κ, and on the matrix P defining the BQLF* (10.18)*, are in effect at all the vertices $\mathscr{V}(\tilde{\mathscr{X}} \times \mathscr{X}_\chi)$:*

$$\forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}} \times \mathscr{X}_\chi),$$
$$\mathsf{P} + \mathsf{L}\Psi_1(\tilde{\mathbf{x}}, \chi) + \Psi_1'(\tilde{\mathbf{x}}, \chi)\mathsf{L}' > 0, \tag{10.34}$$
$$\begin{pmatrix} {\Gamma_1}'{\mathsf{A}_1}'\tilde{\Theta}'(\tilde{\mathbf{x}}, \chi)\mathsf{P} + \mathsf{P}\tilde{\Theta}(\tilde{\mathbf{x}}, \chi)\mathsf{A}_1\Gamma_1 & \star \\ {\mathsf{A}_2}'\tilde{\Theta}'(\tilde{\mathbf{x}}, \chi)\mathsf{P} & \mathsf{O} \end{pmatrix} + \mathsf{W}\Psi_2(\tilde{\mathbf{x}}, \chi) + \Psi_2'(\tilde{\mathbf{x}}, \chi)\mathsf{W}' < 0, \tag{10.35}$$
$$\forall k \in \Xi_K,\ \begin{pmatrix} (2\eta_k - 1) & \star \\ -\eta_k {\Gamma_1}'\mathbf{a}_k & \mathsf{P} \end{pmatrix} + \mathsf{Y}_k\Psi_3(\tilde{\mathbf{x}}, \chi) + \Psi_3'(\tilde{\mathbf{x}}, \chi)\mathsf{Y}_k' \geq 0, \tag{10.36}$$

$$\forall j \in \Xi_J,\ \begin{pmatrix} (2\sigma_j\overline{\zeta}_j - 1) & \star & \star \\ -\sigma_j {\Gamma_1}' \mathbf{K_{1j}} & \mathsf{P} & \mathsf{O} \\ -\sigma_j \mathbf{K_{2j}} & \mathsf{O} & \mathsf{O} \end{pmatrix} + \mathsf{F}_j \Psi_{4,j}(\tilde{\mathbf{x}},\chi) + \Psi'_{4,j}(\tilde{\mathbf{x}},\chi)\mathsf{F}'_j \geq 0, \tag{10.37}$$

$$\forall j \in \Xi_J,\ \begin{pmatrix} -(1+2\tau_j\underline{\zeta}_j) & \star & \star \\ \tau_j {\Gamma_1}' \mathbf{K_{1j}} & \mathsf{P} & \mathsf{O} \\ \tau_j \mathbf{K_{2j}} & \mathsf{O} & \mathsf{O} \end{pmatrix} + \mathsf{G}_j \Psi_{4,j}(\tilde{\mathbf{x}},\chi) + \Psi'_{4,j}(\tilde{\mathbf{x}},\chi)\mathsf{G}'_j \geq 0, \tag{10.38}$$

$$\begin{pmatrix} (1-\kappa) & \star & \star \\ 0 & -\mathsf{P} & \star \\ \mathbf{0} & \mathsf{O} & \kappa\mathsf{I} \end{pmatrix} + \mathsf{K}\Psi_5(\tilde{\mathbf{x}},\chi) + \Psi'_5(\tilde{\mathbf{x}},\chi)\mathsf{K}' \geq 0, \tag{10.39}$$

with

$$\begin{aligned}
\Psi_1(\tilde{\mathbf{x}},\chi) &\triangleq \begin{pmatrix} \mathsf{O} & \Lambda_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}) \\ \mathsf{I}_{n_\Theta} & -\Theta(\tilde{\mathbf{x}},\chi) \end{pmatrix}, \\
\Psi_2(\tilde{\mathbf{x}},\chi) &\triangleq \begin{pmatrix} \Psi_1(\tilde{\mathbf{x}},\chi) & \mathsf{O} \\ \Omega_1(\tilde{\mathbf{x}},\chi)\Gamma_1 & \Omega_2(\tilde{\mathbf{x}},\chi) \end{pmatrix},\ \Lambda_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}) \triangleq \begin{pmatrix} \tilde{\mathbf{x}}_2 & -\tilde{\mathbf{x}}_1 & 0 & \dots & 0 & 0 \\ \tilde{\mathbf{x}}_3 & 0 & -\tilde{\mathbf{x}}_1 & \dots & 0 & 0 \\ \tilde{\mathbf{x}}_{\tilde{n}} & 0 & 0 & \dots & 0 & -\tilde{\mathbf{x}}_1 \\ 0 & \tilde{\mathbf{x}}_3 & -\tilde{\mathbf{x}}_2 & \dots & 0 & 0 \\ \vdots & \vdots & & & & \vdots \\ 0 & \tilde{\mathbf{x}}_{\tilde{n}} & 0 & \dots & 0 & -\tilde{\mathbf{x}}_2 \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & 0 & \dots & \tilde{\mathbf{x}}_{\tilde{n}} & -\tilde{\mathbf{x}}_{(\tilde{n}-1)} \end{pmatrix}, \\
\Psi_3(\tilde{\mathbf{x}},\chi) &\triangleq \begin{pmatrix} \tilde{\mathbf{x}} & -\Gamma_1 \\ \mathbf{0} & \Psi_1(\tilde{\mathbf{x}},\chi) \end{pmatrix}, \\
\Psi_{4,j}(\tilde{\mathbf{x}},\chi) &\triangleq \begin{pmatrix} \Psi_3(\tilde{\mathbf{x}},\chi) & \mathsf{O} \\ (\mathbf{0}\ \Upsilon_{1j}(\tilde{\mathbf{x}},\chi)\Gamma_1) & \Upsilon_{2j}(\tilde{\mathbf{x}},\chi) \end{pmatrix}, \\
\Psi_5(\tilde{\mathbf{x}},\chi) &\triangleq \begin{pmatrix} -\hat{\tilde{\mathbf{x}}}_0 & \Gamma_1 & -\mathsf{E} \\ \mathbf{0} & \Psi_1(\tilde{\mathbf{x}},\chi) & \mathsf{O} \end{pmatrix}, \qquad \textit{and}\ \ \tilde{\mathbf{x}} = (\,\tilde{\mathbf{x}}_1\ \dots\ \tilde{\mathbf{x}}_{\tilde{n}}\,)'.
\end{aligned} \tag{10.40}$$

Recall that a criterion to be optimized under the above set of LMIs can be introduced along the lines of Section 10.3.1.2.

Proof. From the developments of Section 10.4.2.1, the properties underlying the definition of the multicriteria basin of attraction $\tilde{\mathscr{E}}$ have been turned into inequalities of the form (10.14), which depend quadratically on the vector functions $\phi_1(\tilde{\mathbf{x}},\chi)$, $\phi_2(\tilde{\mathbf{x}},\chi)$, …, $\phi_5(\tilde{\mathbf{x}},\chi)$. Moreover, the matrix functions defined in (10.40) depend affinely on their arguments and satisfy

$$\forall(\tilde{\mathbf{x}},\chi) \in \tilde{\mathscr{X}}\times\mathscr{X}_\chi,\ \Lambda_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}) = \mathbf{0}; \qquad \forall l \in \Xi_5,\ \Psi_l(\tilde{\mathbf{x}},\chi)\phi_l(\tilde{\mathbf{x}},\chi) = \mathbf{0}. \tag{10.41}$$

So, $\Lambda_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$ and $\Psi_l(\tilde{\mathbf{x}},\chi)$, $l \in \Xi_5$, are linear annihilators of $\tilde{\mathbf{x}}$ and $\phi_l(\tilde{\mathbf{x}},\chi)$, respectively. The LMIs (10.34)–(10.39) on the matrix P entailed in the definition of $V_\mathsf{P}(\tilde{\mathbf{x}},\chi)$ and on other decision variables straightly follow from the application of Lemma 10.2 to (10.23)–(10.33). Importantly, the conservativeness of these sufficient conclusions similar to (10.15) can be all the more reduced as the number of lines of $\phi_l(.,.)$, $l \in \Xi_5$, is important. This is why $\Lambda_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}})$ appears in $\phi_l(.,.)$, $l \in \Xi_5$. □

10.4.3 LMI Conditions for Multicriteria Analysis Based on PW-BQLFs

The counterparts of the results developed in Section 10.4.2 when using piecewise biquadratic Lyapunov functions are hereafter sketched out. In addition to the above

definitions of Γ_1 and $\mathbf{a}_k$, $k \in \Xi_K$, let each convex cell $\tilde{\mathscr{X}}_s$ of $\tilde{\mathscr{X}}$, $s \in \Xi_S$, be characterized by $n_{\mathbf{v}(s)}$ vectors $\mathbf{v}_{(s,k_\mathbf{v})}$, $k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}}$, and $n_{\mathbf{z}(s)}$ vectors $\mathbf{z}_{(s,k_\mathbf{z})}$, $k_\mathbf{z} \in \Xi_{n_{\mathbf{z}(s)}}$:

$$\tilde{\mathscr{X}}_s = \{\tilde{\mathbf{x}} \in \mathbb{R}^{\tilde{n}} : \mathbf{v}'_{(s,k_\mathbf{v})}\tilde{\mathbf{x}} \leq 1,\ \mathbf{z}'_{(s,k_\mathbf{z})}\tilde{\mathbf{x}} \leq 0,\ k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}},\ k_\mathbf{z} \in \Xi_{n_{\mathbf{z}(s)}}\}. \tag{10.42}$$

Let the vectors $\mathbf{e}_{s_a s_b}$ "support" the boundaries $\partial\tilde{\mathscr{X}}_{s_a s_b}$ of adjacent polytopes, *i.e.*

$$\forall (s_a, s_b) \in \Pi,\ \partial\tilde{\mathscr{X}}_{s_a s_b} \triangleq \tilde{\mathscr{X}}_{s_a} \cap \tilde{\mathscr{X}}_{s_b} \subset \{\tilde{\mathbf{x}} : \mathbf{e}'_{s_a s_b}\tilde{\mathbf{x}} = 0\}, \tag{10.43}$$

with Π as in (10.19). Define a constant matrix $\overline{\mathsf{E}}_{s_a s_b} \in \mathbb{R}^{(n_\Theta+\tilde{n})\times(n_\Theta+\tilde{n})}$ such that

$$\forall (\tilde{\mathbf{x}}, \chi) \in \partial\tilde{\mathscr{X}}_{s_a s_b} \times \mathscr{X}_\chi,\ \overline{\mathsf{E}}_{s_a s_b}\phi_1(\tilde{\mathbf{x}}, \chi) = \mathbf{0}. \tag{10.44}$$

A simple choice of $\overline{\mathsf{E}}_{s_a s_b}$ is $\overline{\mathsf{E}}_{s_a s_b} = {\Gamma_1}'\mathbf{e}_{s_a s_b}\mathbf{e}'_{s_a s_b}\Gamma_1$. However, less conservative results can be obtained for other definitions of $\overline{\mathsf{E}}_{s_a s_b}$ depending on the choice of the partition and the matrix $\Theta(\tilde{\mathbf{x}}, \chi)$, see end of Section 10.5.

The following steps enable the definition of LMI sufficient conditions for $\tilde{\mathscr{E}}$ defined within Definition 10.6 to be a multicriteria basin of attraction for (10.4) subject to (10.5):

- $\forall s \in \Xi_S, \forall (\tilde{\mathbf{x}}, \chi) \in \big(\tilde{\mathscr{X}}_s \backslash \{\mathbf{0}\}\big) \times \mathscr{X}_\chi,\ V_s(\tilde{\mathbf{x}}, \chi) > 0;$
- $\forall s \in \Xi_S, \forall (\tilde{\mathbf{x}}, \chi) \in \big(\tilde{\mathscr{X}}_s \backslash \{\mathbf{0}\}\big) \times \mathscr{X}_\chi,\ \dot{V}_s(\tilde{\mathbf{x}}, \chi) < 0;$
- $\forall s \in \Xi_S, \forall k \in \Xi_K, \forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}} \times \mathscr{X}_\chi,\ (V_s(\tilde{\mathbf{x}}, \chi) \leq 1) \Rightarrow \big(\mathbf{a}'_k\tilde{\mathbf{x}} \leq 1\big);$
- $\forall s \in \Xi_S, \forall j \in \Xi_J, \forall (\tilde{\mathbf{x}}, \chi) \in \tilde{\mathscr{X}}_s \times \mathscr{X}_\chi,$

$$\begin{Bmatrix} V_s(\tilde{\mathbf{x}}, \chi) \leq 1 \\ \mathbf{v}'_{(s,k_\mathbf{v})}\tilde{\mathbf{x}} \leq 1,\ \forall k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}} \\ \mathbf{z}'_{(s,k_\mathbf{z})}\tilde{\mathbf{x}} \leq 0,\ \forall k_\mathbf{z} \in \Xi_{n_{\mathbf{z}(s)}} \end{Bmatrix} \Rightarrow \begin{Bmatrix} {\mathbf{Z}_j}'(\tilde{\mathbf{x}})\tilde{\mathbf{x}} \leq \overline{\zeta}_j; \\ {\mathbf{Z}_j}'(\tilde{\mathbf{x}})\tilde{\mathbf{x}} \geq \underline{\zeta}_j; \end{Bmatrix}$$

- $\forall (s_a, s_b) \in \Pi,\ \forall (\tilde{\mathbf{x}}, \chi) \in \partial\tilde{\mathscr{X}}_{s_a s_b} \times \mathscr{X}_\chi,\ V_{s_a}(\tilde{\mathbf{x}}, \chi) = V_{s_b}(\tilde{\mathbf{x}}, \chi);$
- $\forall (s_a, s_b) \in \Pi,\ \forall (\tilde{\mathbf{x}}, \chi) \in \partial\tilde{\mathscr{X}}_{s_a s_b} \times \mathscr{X}_\chi,\ \dot{V}_{s_a}(\tilde{\mathbf{x}}, \chi) = \dot{V}_{s_b}(\tilde{\mathbf{x}}, \chi).$

Theorem 10.2 (Multicriteria Analysis via PW-BQLFs). *$\tilde{\mathscr{E}}$ defined within Definition 10.6 is a multicriteria basin of attraction for the visual servo* (10.4) *subject to the constraints* (10.5) *if the following LMIs on the matrices* $\{\mathsf{P}_s\}_{s\in\Xi_S}$, $\{\mathsf{L}_s\}_{s\in\Xi_S}$, $\{\mathsf{W}_s\}_{s\in\Xi_S}$, $\{\mathsf{Y}_{s,k}\}_{s\in\Xi_S, k\in\Xi_K}$, $\{\mathsf{F}_{s,j}\}_{s\in\Xi_S, j\in\Xi_J}$, $\{\mathsf{G}_{s,j}\}_{s\in\Xi_S, j\in\Xi_J}$, $\{\mathsf{M}_{s_a s_b}\}_{(s_a,s_b)\in\Pi}$, *and* $\{\mathsf{N}_{s_a s_b}\}_{(s_a,s_b)\in\Pi}$, *on the positive scalars* $\{\eta_{s,k}\}_{s\in\Xi_S, k\in\Xi_K}$, $\{\sigma_{0(s,j)}\}_{s\in\Xi_S, j\in\Xi_J}$, *and* $\{\tau_{0(s,j)}\}_{s\in\Xi_S, j\in\Xi_J}$, *and on the nonnegative scalars* $\{\sigma_{\mathbf{v}(s,j,k_\mathbf{v})}\}_{s\in\Xi_S, j\in\Xi_J, k_\mathbf{v}\in\Xi_{n_{\mathbf{v}(s)}}}$, $\{\sigma_{\mathbf{z}(s,j,k_\mathbf{z})}\}_{s\in\Xi_S, j\in\Xi_J, k_\mathbf{z}\in\Xi_{n_{\mathbf{z}(s)}}}$, $\{\tau_{\mathbf{v}(s,j,k_\mathbf{v})}\}_{s\in\Xi_S, j\in\Xi_J, k_\mathbf{v}\in\Xi_{n_{\mathbf{v}(s)}}}$, $\{\tau_{\mathbf{z}(s,j,k_\mathbf{z})}\}_{s\in\Xi_S, j\in\Xi_J, k_\mathbf{z}\in\Xi_{n_{\mathbf{z}(s)}}}$, *are in effect:*

$$\forall s \in \Xi_S, \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}}_s \times \mathscr{X}_\chi),\ \mathsf{P}_s + \mathsf{L}_s\Psi_1(\tilde{\mathbf{x}}, \chi) + \Psi'_1(\tilde{\mathbf{x}}, \chi)\mathsf{L}'_s > 0, \tag{10.45}$$

$$\forall s \in \Xi_S, \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}}_s \times \mathscr{X}_\chi),$$
$$\mathscr{M}_{(\tilde{\mathbf{x}},\chi)}(\mathsf{P}_s) + \mathsf{W}_s\Psi_2(\tilde{\mathbf{x}}, \chi) + \Psi'_2(\tilde{\mathbf{x}}, \chi)\mathsf{W}'_s < 0, \tag{10.46}$$

$$\forall s \in \Xi_S,\ \forall k \in \Xi_K,\ \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}} \times \mathscr{X}_\chi),$$
$$\begin{pmatrix} (2\eta_{s,k}-1) & \star \\ -\eta_{s,k}{\Gamma_1}'\mathbf{a}_k & \mathsf{P}_s \end{pmatrix} + \mathsf{Y}_{s,k}\Psi_3(\tilde{\mathbf{x}}, \chi) + \Psi'_3(\tilde{\mathbf{x}}, \chi)\mathsf{Y}'_{s,k} \geq 0, \tag{10.47}$$
$$\forall s \in \Xi_S,\ \forall j \in \Xi_J,\ \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}_s} \times \mathscr{X}_\chi),$$
$$\mathscr{Q}_{(\tilde{\mathbf{x}},\chi)}(P_s, \ldots) + \mathsf{F}_{s,j}\Psi_{4,j}(\tilde{\mathbf{x}}, \chi) + \Psi'_{4,j}(\tilde{\mathbf{x}}, \chi)\mathsf{F}'_{s,j} \geq 0, \tag{10.48}$$
$$\forall s \in \Xi_S,\ \forall j \in \Xi_J,\ \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\tilde{\mathscr{X}_s} \times \mathscr{X}_\chi),$$
$$\mathscr{R}_{(\tilde{\mathbf{x}},\chi)}(P_s, \ldots) + \mathsf{G}_{s,j}\Psi_{4,j}(\tilde{\mathbf{x}}, \chi) + \Psi'_{4,j}(\tilde{\mathbf{x}}, \chi)\mathsf{G}'_{s,j} \geq 0, \tag{10.49}$$

$$\forall (s_a, s_b) \in \Pi,\ \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\partial\tilde{\mathscr{X}}_{s_a s_b} \times \mathscr{X}_\chi)$$
$$\widehat{\overline{\mathsf{E}}_{s_a s_b}}' \left(P_{s_a} - P_{s_b} + \mathsf{M}_{s_a s_b}\Psi_1(\tilde{\mathbf{x}}, \chi) + \Psi'_1(\tilde{\mathbf{x}}, \chi)\mathsf{M}'_{s_a s_b} \right) \widehat{\overline{\mathsf{E}}_{s_a s_b}} = \mathsf{O}, \tag{10.50}$$
$$\forall (s_a, s_b) \in \Pi,\ \forall (\tilde{\mathbf{x}}, \chi) \in \mathscr{V}(\partial\tilde{\mathscr{X}}_{s_a s_b} \times \mathscr{X}_\chi)$$
$$\widehat{\overline{\overline{\mathsf{E}}}_{s_a s_b}}' \left(\mathscr{M}_{(\tilde{\mathbf{x}},\chi)}(\mathsf{P}_{s_a} - \mathsf{P}_{s_b}) + \mathsf{N}_{s_a s_b}\Psi_2(\tilde{\mathbf{x}}, \chi) + \Psi'_2(\tilde{\mathbf{x}}, \chi)\mathsf{N}'_{s_a s_b} \right) \widehat{\overline{\overline{\mathsf{E}}}_{s_a s_b}} = \mathsf{O}, \tag{10.51}$$

with:

- $\mathscr{M}_{(\tilde{\mathbf{x}},\chi)}(\mathsf{P}_s) \triangleq \begin{pmatrix} {\Gamma_1}'{\mathsf{A}_1}'\tilde{\Theta}'(\tilde{\mathbf{x}},\chi)\mathsf{P}_s + \mathsf{P}_s\tilde{\Theta}(\tilde{\mathbf{x}},\chi)\mathsf{A}_1\Gamma_1 & \star \\ {\mathsf{A}_2}'\tilde{\Theta}'(\tilde{\mathbf{x}},\chi)\mathsf{P}_s & \mathsf{O} \end{pmatrix}$,
$$\mathscr{Q}_{(\tilde{\mathbf{x}},\chi)}(P_s, \ldots) \triangleq \begin{pmatrix} (2\sigma_{0(s,j)}\overline{\zeta}_j - 1 - 2\sum_{k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}}} \sigma_{\mathbf{v}(s,j,k_\mathbf{v})}) & \star & \star \\ {\Gamma_1}'\left(-\sigma_{0(s,j)}\mathbf{K_{1j}} + \sum_{k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}}} \sigma_{\mathbf{v}(s,j,k_\mathbf{v})}\mathbf{v}_{(s,k_\mathbf{v})} + \sum_{k_\mathbf{z} \in \Xi_{n_{\mathbf{z}(s)}}} \sigma_{\mathbf{z}(s,j,k_\mathbf{z})}\mathbf{z}_{(s,k_\mathbf{z})}\right) & \mathsf{P}_s & \mathsf{O} \\ -\sigma_{0(s,j)}\mathbf{K_{2j}} & \mathsf{O} & \mathsf{O} \end{pmatrix},$$
$$\mathscr{R}_{(\tilde{\mathbf{x}},\chi)}(P_s, \ldots) \triangleq \begin{pmatrix} (-2\tau_{0(s,j)}\underline{\zeta}_j - 1 - 2\sum_{k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}}} \tau_{\mathbf{v}(s,j,k_\mathbf{v})}) & \star & \star \\ {\Gamma_1}'\left(\tau_{0(s,j)}\mathbf{K_{1j}} + \sum_{k_\mathbf{v} \in \Xi_{n_{\mathbf{v}(s)}}} \tau_{\mathbf{v}(s,j,k_\mathbf{v})}\mathbf{v}_{(s,k_\mathbf{v})} + \sum_{k_\mathbf{z} \in \Xi_{n_{\mathbf{z}(s)}}} \tau_{\mathbf{z}(s,j,k_\mathbf{z})}\mathbf{z}_{(s,k_\mathbf{z})}\right) & \mathsf{P}_s & \mathsf{O} \\ \tau_{0(s,j)}\mathbf{K_{2j}} & \mathsf{O} & \mathsf{O} \end{pmatrix};$$
- $\widehat{\overline{\mathsf{E}}_{s_a s_b}} \in \mathbb{R}^{(n_\Theta + \tilde{n}) \times m_{\overline{\mathsf{E}}}}$ *a full-rank matrix spanning the nullspace of* $\overline{\mathsf{E}}_{s_a s_b}$*, i.e. such that* $\overline{\mathsf{E}}_{s_a s_b}\widehat{\overline{\mathsf{E}}_{s_a s_b}} = \mathsf{O}$ *and* $\mathrm{rank}(\widehat{\overline{\mathsf{E}}_{s_a s_b}}) = m_{\overline{\mathsf{E}}}$*;*
- $\overline{\overline{\mathsf{E}}}_{s_a s_b} \triangleq \begin{pmatrix} \overline{\mathsf{E}}_{s_a s_b} & \mathsf{O} \\ \mathsf{O} & \mathsf{O}_{n\pi} \end{pmatrix}$*;* $\widehat{\overline{\overline{\mathsf{E}}}_{s_a s_b}}$ *full-rank with* $\overline{\overline{\mathsf{E}}}_{s_a s_b}\widehat{\overline{\overline{\mathsf{E}}}_{s_a s_b}} = \mathsf{O}$*,* e.g. $\widehat{\overline{\overline{\mathsf{E}}}_{s_a s_b}} \triangleq \begin{pmatrix} \widehat{\overline{\mathsf{E}}_{s_a s_b}} & \mathsf{O} \\ \mathsf{O} & \mathsf{I}_{n\pi} \end{pmatrix}$*;*
- $\Psi_1(.,.)$*,* $\Psi_2(.,.)$*,* $\Psi_3(.,.)$*,* $\Psi_{4,j}(.,.)$*,* $\Psi_5(.,.)$ *and* $\Lambda_{\tilde{\mathbf{x}}}(.)$ *as defined in* (10.40)*.*

Proof. The proof is not included for space reasons, but follows the lines of the proof of Theorem 10.1 and of [9]. Note that the conservativeness of (10.47) can be reduced. □

A multicriteria basin of attraction $\tilde{\mathscr{E}}$ satisfying (10.45)–(10.51) can be expanded in many ways, *e.g.* through the heuristic maximization of its extent towards a set of selected points as suggested in Section 10.3.1.2, or by defining an ellipsoid (10.31) included in $\tilde{\mathscr{E}}$ and maximizing a function of its shape matrix E related to its "size".

10.5 Case Study

The image-based positioning of a 2 DOF camera with respect to a still target is considered (Figure 10.2). The camera can move along and around its optical axis, so that $\mathbf{u}$ and $\mathbf{x}$ write as $\mathbf{u} = \begin{pmatrix} v_z & \omega_z \end{pmatrix}'$ and $\mathbf{x} = \begin{pmatrix} t_z & N \triangleq \tan(\frac{\nu}{4}) \end{pmatrix}'$. Its focal length is normalized to $f = 1$. The target is fitted with two asymmetric spots T_1, T_2 characterized by $\overrightarrow{z_S} = \overrightarrow{z_T}$, $a_1 = a_2 = 0$, $b_1 = 1$, $b_2 = -2$, $c_1 = c_2 = c = 1.5$. The visual feedback $\mathbf{u} = -\lambda [\mathsf{J}(\mathbf{s}^*, \mathbf{z}^*)]^+ (\mathbf{s} - \mathbf{s}^*)$ described in [14, 3] is implemented, with $\lambda = 0.1$, $\mathbf{z}^* = \begin{pmatrix} c & c \end{pmatrix}'$ and $\mathsf{J}(\mathbf{s}^*, \mathbf{z}^*)$ the interaction matrix computed at the reference situation.

As the controller is static, $\tilde{\mathbf{x}} = \mathbf{x}$ holds, and the closed-loop state equation writes as

$$\begin{pmatrix} \dot{t}_z \\ \dot{N} \end{pmatrix} = \frac{\lambda}{t_z + c} \begin{pmatrix} -c & \frac{-8c^2 N}{(1+N^2)^2} \\ 0 & \frac{-c(1-N^2)}{(1+N^2)} \end{pmatrix} \begin{pmatrix} t_z \\ N \end{pmatrix}. \tag{10.52}$$

No 3D constraint is imposed on the motion. The actuators limits are $|v_z| \leq 1.5\,\mathrm{m.s}^{-1}$ and $|\omega_z| \leq 1\,\mathrm{rad.s}^{-1}$. For $f = 1$, the virtual limits of the image plane are set to $|\overline{x}_i| \leq 4\,\mathrm{m}$ and $|\overline{y}_i| \leq 3\,\mathrm{m}$. These constraints are expressed thanks to the following expressions of $\mathbf{y} = \mathbf{s} - \mathbf{s}^* = \mathsf{C}(\tilde{\mathbf{x}})\tilde{\mathbf{x}}$ and of the controller gain $\mathsf{K} = -\lambda [\mathsf{J}(\mathbf{s}^*, \mathbf{z}^*)]^+$:

$$\begin{pmatrix} \overline{x}_i - \overline{x}_i^* \\ \overline{y}_i - \overline{y}_i^* \end{pmatrix} = \frac{1}{t_z + c_i} \begin{pmatrix} -\frac{a_i}{c_i} & \frac{4b_i N^2 - 8a_i N - 4b_i}{(1+N^2)^2} \\ -\frac{b_i}{c_i} & \frac{-4a_i N^2 - 8b_i N + 4a_i}{(1+N^2)^2} \end{pmatrix} \begin{pmatrix} t_z \\ N \end{pmatrix}; \qquad \mathsf{K} = \frac{-\lambda c \begin{pmatrix} ca_1 & cb_1 & ca_2 & cb_2 \\ b_1 & -a_1 & b_2 & -a_2 \end{pmatrix}}{a_1^2 + b_1^2 + a_2^2 + b_2^2}. \tag{10.53}$$

As per [3], the convergence fails if $N = 1$ at initial time. Note that despite its apparent simplicity, the exact *multicriteria* basin of attraction of this 2D servo is unknown.

First, $\tilde{\mathscr{E}}$ is defined as the union $\cup_{r \in \Xi_R} (\tilde{\mathscr{E}})_r$ of elementary multicriteria basins of attraction computed through separate optimization problems as suggested in Section 10.3.1.3. BQLFs of the form $V(\tilde{\mathbf{x}}) = (t_z^2, t_z N, N^2, t_z, N)' \mathsf{P} (t_z^2, t_z N, N^2, t_z, N)$ are considered, *i.e.* the matrix function $\Theta(\tilde{\mathbf{x}}, \chi)$ and $\tilde{\Theta}(\tilde{\mathbf{x}}, \chi)$ are set to $\Theta(\tilde{\mathbf{x}}, \chi) = \Theta(\tilde{\mathbf{x}}) = \begin{pmatrix} t_z \mathrm{I}_2 & N \mathrm{I}_2 \end{pmatrix}'$ and $\tilde{\Theta}(\tilde{\mathbf{x}}, \chi) = \tilde{\Theta}(\tilde{\mathbf{x}}) = \begin{pmatrix} 2t_z & N & N & 0 \\ 0 & t_z & t_z & 2N \end{pmatrix}'$. The results are shown on Figure 10.3, endowed with an horizontal t_z-axis and a vertical N-axis. The admissible subset of the state space is the area on the right of the left vertical curves, defined from the actuators and visibility constraints.

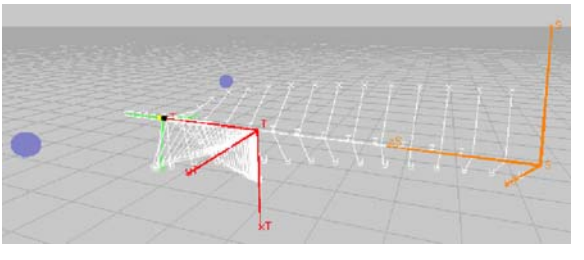


Fig. 10.2 2 DOF image-based control

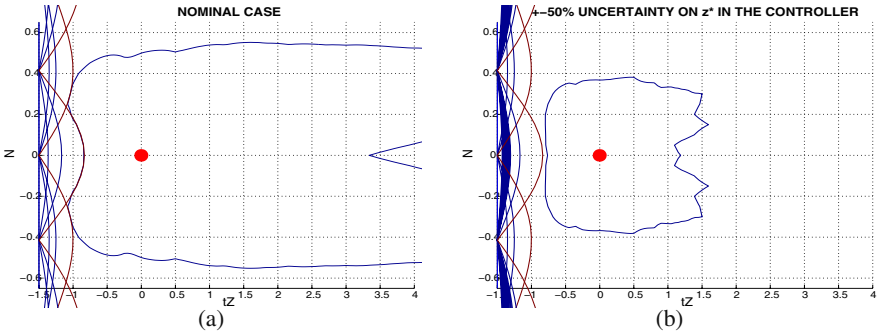


Fig. 10.3 Multicriteria analysis of a 2 DOF image-based servo [12]: (a) nominal case; and (b) $\pm 50\%$ uncertainty on $\mathbf{z}^*$. The horizontal and vertical axes correspond to the relative sensor-target translation and rotation. The point $(0,0)$ corresponds to the achievement of the positioning task. The admissible sensor-target situations are on the right of the left vertical curves, which portray the actuators and visibility limits. Contrarily to (a), in (b) several curves express the constraint on v_z, which depends on χ

Very often, *e.g.* when the visual servoing task is defined from the image data only, the depth $\mathbf{z}^*$ of the target spots at the reference situation can just be approximated up to a bounded constant uncertainty χ by $\widehat{\mathbf{z}^*} = \mathbf{z}^*(1+\chi)$. Building the control law with $[\mathsf{J}(\mathbf{s}^*, \widehat{\mathbf{z}^*})]$ instead of $[\mathsf{J}(\mathbf{s}^*, \mathbf{z}^*)]$ may cause serious convergence problems [18]. Simulations considering $\chi \in [-0.5; +0.5]$ and $\dot{\chi} = 0$ are reported on Figure 10.4. To limit the conservativeness, the considered biquadratic Lyapunov functions depend on the uncertainty χ through the matrix function $\Theta(.,.)$. Here, $V(\tilde{\mathbf{x}}, \chi)$ writes as $V(\tilde{\mathbf{x}}, \chi) = (t_z^2, t_z N, N^2, t_z, N, \chi)' \mathsf{P} (t_z^2, t_z N, N^2, t_z, N, \chi)$.

Even if the extents of the elementary basins composing $\tilde{\mathscr{E}}$ in Figure 10.3 have not been intensively optimized, BQLFs seem fairly limited. Figure 4(a) shows that BQLFs do not provide enough DOF in order to expand the consequent multicriteria

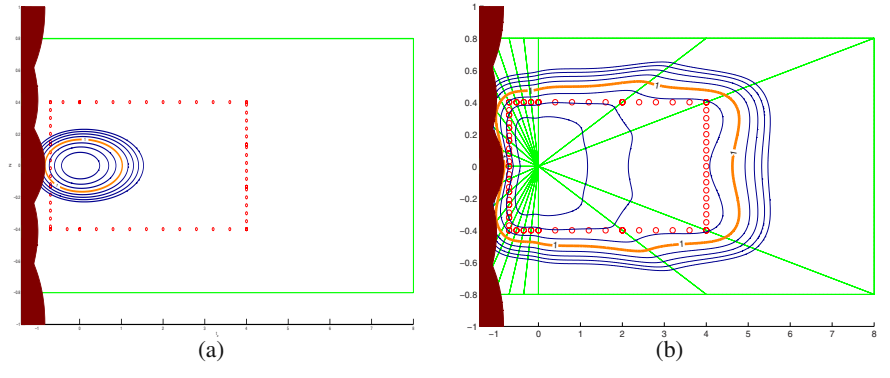


Fig. 10.4 2D servo multicriteria basin of attraction extended towards all the circled points: (a) obtained from a single BQLF; and (b) obtained from a PW-BQLF

basin of attraction (the thickest 1-level set) towards several distinct directions simultaneously. Figure 4(b) suggests that PW-BQLFs constitute a promising alternative, still at a reasonable complexity. Note that in Figures 4(a)–4(b), if $\mathbf{e}_{s_a s_b} = \begin{pmatrix} v_1 & v_2 \end{pmatrix}'$, then $\overline{\mathsf{E}}_{s_a s_b}$ reads as

$$\overline{\mathsf{E}}_{s_a s_b} = \begin{pmatrix} 0 & 0 & 0 & 0 & v_1 & v_2 \\ v_1 & v_2 & 0 & 0 & 0 & 0 \\ v_1 & 0 & v_2 & 0 & 0 & 0 \\ 0 & v_1 & 0 & v_2 & 0 & 0 \\ 0 & 0 & v_1 & v_2 & 0 & 0 \end{pmatrix}.$$

10.6 Conclusion

A new framework has been proposed to the analysis of a large class of 3D and 2D eye-in-hand visual servos. Aside from convergence, various criteria (*e.g.* actuators, 2D or 3D constraints) can be handled. The genuine problem has been turned into the analysis of a nonlinear rational system subject to rational constraints. By using DARs and extensions of the S-procedure, LMI conditions could be obtained, enabling the definition of a "multicriteria basin of attraction" as a level set of a BQLF. Though the consequent optimization programs are convex and of polynomial complexity, the total computational cost gets fairly important if one makes the union of basins of attraction computed separately. The extension to PW-BQLFs was shown to lead to tolerably conservative conclusions still at a reasonable computational cost.

Short-term incremental developments will concern an evaluation of the method on more complex case studies, and its comparison with solutions based on other types of Lyapunov functions, *e.g.* polyquadratic Lyapunov functions, or homogeneous polynomial Lyapunov functions in the vein of [5]. Multicriteria synthesis will be considered as well, and actuators saturations will be tolerated rather than avoided, which is meaningful in a kinematic context. Last, following the duality acknowledged in [1], a new solution to visual-based localization will be assessed.

More fundamental mean-term objective are twofold. On a control theory aspect, more efficient relaxations will be envisaged so as to get less conservative LMI conditions. On the robotics side, the aim will be to "bridge the gap" between the suggested approach and existing developments in visual servoing. For instance, other visual features will be inspected. One challenge is to get a rational matrix function $\mathsf{C}(.,.)$ of the open-loop rational model (10.2) in closed-form, as has been the case for spotted visual features, which leads to a better conditioning of the problem. The aim is to limit the conservativeness of the conclusions, so that an LMI analysis or synthesis strategy can provide an answer to difficult robotics problems.

Acknowledgements. The first author would like to thank Prof. Alexandre Trofino for early fruitful discussions, and Prof. Michel Courdesses for his comments on the draft of this work.

References

[1] Bellot, D., Danès, P.: An LMI solution to visual-based localization as the dual of visual servoing. In: IEEE Conference on Decision and Control (CDC 2003), Maui, HI, pp. 5420–5425 (2003)

[2] Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

[3] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: The confluence of Vision and Control. LNCIS, vol. 237, pp. 66–78. Springer, Heidelberg (1998)

[4] Chaumette, F., Hutchinson, S.: Visual servo control, part I: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)

[5] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Robust Analysis of LFR Systems Through Homogeneous Polynomial Lyapunov Functions. IEEE Transactions on Automatic Control 49(7), 1211–1216 (2004)

[6] Chesi, G., Hashimoto, K., Prattichizzo, D., Vicino, A.: Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Transactions on Robotics 20(5), 908–914 (2004)

[7] Corke, P., Hutchinson, S.: A new partitioned approach to image-based visual servo control. IEEE Trans. Robotics and Automation 17(4), 507–515 (2001)

[8] Coutinho, D., Bazanella, A., Trofino, A., Silva, A.: Stability Analysis and Control of a Class of Differential-Algebraic Nonlinear Systems. International Journal of Robust and Nonlinear Control 14, 1301–1326 (2004)

[9] Coutinho, D., Danès, P.: Piecewise Lyapunov functions to the stability analysis of rational systems subject to multiple state constraints. In: IEEE Conference on Decision and Control (CDC 2006), San Diego, CA, pp. 5801–5806 (2006)

[10] Cowan, N., Weingarten, J., Koditschek, D.: Visual servoing via navigation functions. IEEE Transactions on Robotics and Automation 18(4), 521–533 (2002)

[11] Danès, P., Bellot, D.: Towards an LMI approach to multicriteria visual servoing in robotics. European Journal of Control 12(1), 86–110 (2006), Companion report: `http://www.laas.fr/~danes/RESEARCH/VS/companionEJC2k6.pdf`

[12] Danès, P., Coutinho, D.F.: Multicriteria analysis of visual servos via biquadratic Lyapunov functions. In: IFAC Symposium on Robust Control Design (ROCOND 2006), Toulouse (2006)

[13] El Ghaoui, L., Scorletti, G.: Control of rational systems using linear-fractional representations and linear matrix inequalities. Automatica 32(9), 1273–1284 (1996)

[14] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation 8(3), 313–326 (1992)

[15] Favrholdt, L.P.E., Paulin, P., Petersen, M., Robust, H.: control for high speed visual servoing applications. International Journal of Advanced Robotic Systems 4(3), 279–292 (2007)

[16] Luo, Z.: Applications of convex optimization in signal processing and digital communication. Mathematical Programming Ser. B 97, 177–207 (2003)

[17] Malis, E., Chaumette, F.: Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. IEEE Transactions on Robotics and Automation 18(2), 176–186 (2002)

[18] Malis, E., Rives, P.: Robustness of image-based visual servoing with respect to depth distribution error. In: IEEE International Conference on Robotics and Automation (ICRA 2003), Taipei, Taiwan, pp. 1056–1061 (2003)

[19] Mansard, N., Chaumette, F.: A new redundancy formalism for avoidance in visual servoing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Canada, pp. 1694–1700 (2005)
[20] Mezouar, Y., Chaumette, F.: Optimal camera trajectory with image-based control. International Journal of Robotics Research 22(10/11) (2003)
[21] Samson, C., Le Borgne, M., Espiau, B.: Robot Control: The Task Function Approach. Clarendon Press, Oxford (1991)
[22] Schramm, F., Morel, G., Micaelli, A., Lottin, A.: Extended-2D visual servoing. In: IEEE International Conference on Robotics and Automation (ICRA 2004), New Orleans, LA, pp. 267–273 (2004)
[23] Souères, P., Tarbouriech, S., Gao, B.: A robust vision-based controller for mobile robots navigation: application to the task sequencing problem. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005), Edmonton, Canada, pp. 2191–2196 (2005)
[24] Trofino, A.: Local, Regional and Global Stability: An LMI Approach for Uncertain Nonlinear Systems, Tech. Report, DAS-CTC, Univ. Federal de Santa Catarina, Brazil (2002) (available under request)
[25] Zhou, K., Doyle, J., Glover, K.: Robust and Optimal Control, Englewood Cliffs (1995)

Chapter 11
Path-Planning for Visual Servoing: A Review and Issues

Moslem Kazemi, Kamal Gupta, and Mehran Mehrandezh

Abstract. In this survey we provide a comprehensive technical review of existing major approaches to path-planning for visual servoing. Visual servoing has been introduced as a promising approach for sensor-based robotic tasks. The basic visual servoing task is to guide the motion of a robot with respect to a target object based on the feedback obtained through a vision system. Amalgamation of path-planning techniques with reactive visual servoing strategies can robustify existing image-based tracking systems in robotics applications where a high disparity between the initial and desired views of a target is inevitable (*e.g.*, target interception, space docking, reaching and grasping, *etc*). The planning stage does so by accounting for critical constraints and uncertainties in the system resulting in a more robust visual servoing process. We discuss different planning approaches, explain the associated set of constraints and assumptions, and discuss the underlying path-planning techniques along with the issues regarding their integration with reactive visual servo controllers.

11.1 Introduction

The role of vision as a sensor for autonomous machines to interact with complex, unknown, and dynamic environments is paramount. Visual servoing has been introduced as a promising approach for sensor-based robotic tasks such as positioning a robot with respect to a target and tracking a moving target via estimating its 3D motion, *i.e.* egomotion analysis using vision. The basic visual servoing task is to

Moslem Kazemi and Kamal Gupta
School of Engineering Science, Simon Fraser University, Burnaby, BC, V5A1S6, Canada
e-mail: {moslemk,kamal}@cs.sfu.ca

Mehran Mehrandezh
Faculty of Engineering and Applied Science, University of Regina, Regina, Saskatchewan S4S0A2
e-mail: mehran.mehrandezh@uregina.ca

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 189–207.
springerlink.com

guide the motion of a robot with respect to a target object based on the feedback obtained through a vision system [31]. Usually an error function $\mathbf{e}$ (also called *task function* [21]) is defined as

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s_d} \tag{11.1}$$

where $\mathbf{s}$ and $\mathbf{s_d}$ denote the vectors of current and desired features, respectively. The visual servoing objective is set to regulate this error to zero.

The existing visual servoing techniques are classified into different categories based on the definition of error function, the underlying control architecture, and the robot-camera configuration (*i.e.*, eye-in-hand vs. eye-to-hand configuration[1]). For a detailed review on existing techniques and their classification see [7], [8], and [31]. In summary, the existing approaches can be classified into two main categories: (1) position-based visual servoing (PBVS) where the feedback is defined in terms of the 3D Cartesian information derived from the image(s), and (2) image-based visual servoing (IBVS) where the feedback is defined directly in the image in terms of image features.

IBVS techniques have better local stability and convergence in presence of camera calibration and modeling errors. However, they suffer from global convergence problems, and, hence, will break down, in particular when the initial and desired camera poses are distant [6]. For example some of the image features might leave the camera's field of view and consequently result in failure of the servoing task. Moreover, there is no direct control on the robot/camera motion induced by the image-based control law. This might result in infeasible maneuvers due to the robot's joint limits and/or collision with workspace obstacles.

Amalgamation of path-planning techniques with reactive image-based visual servoing strategies can robustify existing image based tracking systems by accounting for critical *constraints* and *uncertainties* in robotics applications where a high disparity between the initial and desired views of a target is inevitable (*e.g.*, target interception, space docking, reaching and grasping, *etc*). The main idea of path-planning for visual servoing is to plan and generate *feasible* image trajectories while accounting for certain constraints, and then to servo the robot along the planned trajectories.

In this survey we provide a comprehensive technical review on existing and recent approaches to path-planning for visual servoing. For each approach the set of constraints and the assumptions are explained and the underlying path-planning technique is discussed along with the issues regarding its integration with the reactive image-based controllers.

In Section 11.2 we study the two sets of critical constraints in visual servoing context: (1) image/camera, and (2) robot/physical constraints. The existence of such constraints motivates the need for path-planning techniques aimed at making the servoing process more robust especially in complex visual servoing scenarios. In Section 11.3 a comprehensive overview of the these approaches and their

[1] In an eye-in-hand configuration the camera is mounted on the end-effector of the robot and robot's motion results in camera's motion while in an eye-to-hand configuration, the camera is stationary and looks at the end-effector of the robot and robot's motion does not affect the camera pose [31].

categorization based on the underlying path-planning techniques are provided. In Section 11.4 we discuss the effect of uncertainties on visual servoing and report on some recent works aimed at path-planning under uncertainty for visual servoing. Finally, we conclude the survey in Section 11.5.

11.2 Constraints in Visual Servoing

In [6], through simple, yet, effective examples, Chaumette outlined the potential problems of stability and convergence in both IBVS and PBVS techniques imposed by a number of constraints. Overall one can divide these constraints into two main categories: (1) image/camera, and (2) robot/physical constraints. These two categories are detailed as follows.

11.2.1 Image/Camera Constraints

The image/camera constraints are mainly due to the sensing limits of the vision system or the inter-relationship between the optical flow (*i.e.*, rate of change) of the features $\dot{\mathbf{s}} \in \mathbb{R}^k$ in the image space and the camera's Cartesian velocity $\dot{\mathbf{x}} \in \mathbb{R}^n$ defined through the *image Jacobian* (also called *interaction* matrix) ${}^{\mathbf{s}}\mathbf{L}_{\mathbf{x}} \in \mathbb{R}^{k \times n}$ related to image features [31]:

$$\dot{\mathbf{s}} = {}^{\mathbf{s}}\mathbf{L}_{\mathbf{x}}\dot{\mathbf{x}} \tag{11.2}$$

These constraints are: (1) *field of view limits*, (2) *image local minima*, and (3) *singularities in image Jacobian*.

Field of view limits. The camera as a sensing system have certain limitations. For example the 3D target features projected into the image plane of the camera are visible if their projections fall inside the boundary of the image. The limits of the image are usually represented by a rectangular region which determines the visible region of the image plane. Although in IBVS context the control is directly defined in the image, there is still the possibility that the features leave the camera's field of view, in particular when the initial and desired poses of the camera are distant [6].

Image local minima. As shown in [6], in IBVS context, image local minima might occur due to the existence of unrealizable image motions which do not belong to the range space of image Jacobian ${}^{\mathbf{s}}\mathbf{L}_{\mathbf{x}}$. Hence, there does not exist any camera motion able to produce such unrealizable motions in the image. In general, determining the image local minima is difficult without considering the specific target location and the initial and desired relative camera-target locations, which in turn, leads to an exhaustive search for local minima in the image for each instance of a visual servoing task. As demonstrated in [6], using a nominal value of image Jacobian estimated at the desired location might be of help to avoid local minima in visual servoing tasks. But this may lead to peculiar trajectories of features in the image, which in turn, might violate field of view limits. One should note that the PBVS techniques are known to be free of image local minima since the task function is defined in the Cartesian space.

Singularities in image Jacobian. At these singularities, certain camera motions cannot be achieved by the motion of image features in the image space. Several cases of image singularities have been considered in [6]: the image Jacobian ${}^{\mathbf{s}}\mathbf{L}_{\mathbf{x}}$ is known to be singular if the vector of image features $\mathbf{s}$ consists of the image of (1) three collinear points, or (2) three points belonging to a cylinder containing the camera optical center. Although using more than three noncoplanar points will avoid such singularities, the image Jacobian may still become singular no matter how many feature points (irrespective of their arrangements) are used to define the task function. For example, a visual servoing task involving a 180 degrees rotation around the optical axis results in a singular image Jacobian. As shown in [6] using line features instead of points helps to avoid such singularities, however, it does not completely eliminate the singularities in the image space. *Motion perceptibility* [59] has been proposed as a measure of closeness to image singularities.

11.2.2 Robot/Physical Constraints

Motion of the robot/camera system induced by the visual servo control loop, especially in IBVS, may also violate certain constraints imposed by the robot and/or physical obstacles in the workspace. These are: (1) *robot kinematics* such as joint limits and singularities in robot Jacobian, (2) *robot dynamics*, (3) *collision* with obstacles or self-collision, and (4) *occlusion* due to obstacles, robot body, or self-occlusion by the target.

Over the past three decades a great deal of research in robotics community has been devoted to planning feasible paths avoiding robot kinematics and/or dynamics constraints and collision with physical obstacles or self-collision in various environments (see *e.g.* [39] and [40]). Path-planning approaches have also considered occlusion constraints in applications that require target visibility, *e.g.* [41] and [51].

Since Chaumette's article [6] on the convergence and stability problems of classical visual servoing techniques, most of the efforts in visual servoing community have been devoted to taking the above image/camera and/or robot/physical constraints into account and incorporating them into the reactive visual servoing control loop.

First a number of researchers proposed *partitioned* (or decoupled) control schemes in which certain degrees of freedom are controlled in the manner of IBVS while others are controlled in the manner of PBVS, thereby taking advantage of each individual technique's benefit in avoiding some of the above constraints (see *e.g.* [16], [18], [44], [47], [52]). Each of these partitioned approaches has its own benefits and drawbacks in accounting for the aforementioned constraints. A performance test has been presented in [25] comparing the efficiency of some of the above partitioned strategies with IBVS technique.

Later on, inspired by the theory of hybrid systems, a number of researchers proposed *hybrid* (or switched) strategies consisting a set of visual servo controllers along with a switching rule to switch between them if required (see *e.g.*, [11], [19], [24], [27]). Using switched strategies it might be possible to enlarge the stability

region of classical visual servoing techniques and to switch between a set of unstable controllers to make the overall system stable.

Each of the above partitioned or hybrid strategies deals with only a subset of the above mentioned constraints. Incorporating all the image/camera and robot/physical constraints into the visual servo control loop is, if at all practical, quite challenging. Clearly some sort of path-planning on top of the visual servo control loop is needed for incorporating all the aforementioned constraints, especially in complex visual servoing scenarios.

11.3 Path-planning for Visual Servoing

The main idea of path-planning for visual servoing [49] is to plan and generate *feasible* image trajectories while accounting for the constraints mentioned in the previous section, and then to servo the robot along the planned trajectories. So, the initial large error is discretized and the error to regulate at each cycle of the servoing loop remains small. Overall, this results in a more robust servoing process with respect to the aforementioned image/camera and robot/physical constraints.

Over the past decade research has been devoted to incorporate aspects of path-planning in visual servoing. Although there is no formal classification of the existing path-planning techniques for visual servoing, considering the underlying path-planning approach and the assumptions made in each technique, we divide them into four groups: (1) image space path-planning, (2) optimization-based path-planning, (3) potential field-based path-planning, and (4) global path-planning. In this survey we discuss the major works done in each group to describe the main idea and the underlying problems.

11.3.1 Image Space Path-planning

Image space path-planning techniques aim at interpolating a path in the image space between the initial and desired images without using any knowledge of camera calibration or target model. One of the difficulty of such approaches is that the planned image space path may not correspond to any single path for the camera. So, efforts have been devoted to planning image paths which correspond to feasible (yet unknown) camera paths in an uncalibrated domain. Various results from projective geometry have been applied in this context including: epipolar geometry, projective homography, and projective invariance.

Epipolar geometry. Given multiple views of the same scene, epipolar geometry [26] has been employed by a number of researchers for calibration-free visual servoing. In an early work [29], a trajectory generator for visual servoing was proposed directly in the image space based on epipolar constraints defined between the images obtained from a stereo camera mounted on a robotic arm (eye-to-hand configuration). The task was to accomplish obstacle avoidance (only for robot's end-effector) in an unknown environment. An uncalibrated visual servo controller based

on a Jacobian estimator was used to track the planned image trajectories without using any knowledge of the system or camera calibration.

Park and Chung proposed an image space path-planning approach for an eye-to-hand system using uncalibrated stereo cameras in a vision-based grasping scenario [54]. They generate a number of intermediate views of the robot's gripper along a straight line between the initial image and the final desired image in the projective space with the help of epipolar geometry and without using any 3D information regarding either the gripper or the target object. These intermediate views constitute the desired image trajectories. The robot is then controlled along the image trajectories using the IBVS technique presented in [22]. When followed by the robot, the planned trajectories allow the robot's gripper to track a straight line in the 3D workspace and through out its motion a selected set of features on the gripper are kept in the camera's field of view.

Projective homography. To avoid explicit computation of feasible camera paths which relies on the knowledge of the camera calibration and target model, a number of approaches have been developed using the projective geometry [23] relationship established between the initial and desired images. Working in projective space allows one to partially parameterize the Euclidean displacement of the camera without explicit reconstruction of the Euclidean components.

Projective homography matrix has been employed in the context of path-planning for visual servoing. Projective homography captures the relationship between the images taken from different views of the same scene. Given the projective homogeneous coordinates $\mathbf{p} = (u, v, 1)^T$ and $\mathbf{p}^* = (u^*, v^*, 1)^T$ of a 3D point $\mathbf{P}$ in the current and desired images, respectively, the projective homography matrix $\mathbf{G}$, also called *collineation* matrix, is defined (up to an scale α_g) as

$$\alpha_g \mathbf{p} = \mathbf{G}\mathbf{p}^*. \tag{11.3}$$

The projective homography matrix can be estimated form the knowledge of several features such as points, lines, and contours matched between two images [13], [26], and [46].

In [50] a calibration-free path-planning approach is proposed which consists of interpolating for the *collineation* matrix $\mathbf{G}$ between the initial and desired images to obtain closed-form analytical collineation paths. The image feature trajectories are then derived and followed using an IBVS technique. The proposed approach guarantees convergence to the desired location, however, the convergence does not hold in presence of visibility constraints such as field of view limits. This approach has been extended in [58] to take visibility constraints into account by guiding the image of an arbitrary selected reference point on the target along a straight line in the image which guarantees that the reference point remains in the camera's field of view. However, the camera will not follow a straight line anymore and the other features may still leave camera's field of view. A depth modulation approach has been proposed to keep the visibility of other features by controlling the camera backwards along an optical ray whenever a feature reaches the borders of camera's field of view.

If the camera calibration is known, one could derive further information regarding the camera transformation. In particular, one can compute the Euclidean homography matrix $\mathbf{H}$ (up to a scale α_h) as

$$\alpha_h \mathbf{H} = \mathbf{K}^{-1}\mathbf{G}\mathbf{K} \tag{11.4}$$

where $\mathbf{K}$ is a nonsingular matrix and contains the camera intrinsic parameters. The Euclidean homography (from a set of planar features) can be decomposed to obtain the corresponding (Euclidean) camera transformation parameters as

$$\mathbf{H} \Rightarrow \left\{\mathbf{R}, \frac{\mathbf{t}}{d^*}, \mathbf{n}^*\right\} \tag{11.5}$$

where $\mathbf{R}$ and $\mathbf{t}$ denote the translation and rotation from the desired to the current camera frame, and d^* is the distance of the plane containing the features from the desired camera frame and $\mathbf{n}^*$ is the normal to the plane expressed with respect to the desired frame.

Decomposition of Euclidean homography has been employed by some researchers to plan for image paths corresponding to feasible (yet unknown) camera paths without explicit reconstruction of the camera paths in the Cartesian space.

A shortest path approach has been proposed in [36] which avoids the use of 3D reconstruction by using homography-based partial pose estimation. The proposed approach moves the in-hand camera directly along the direction (obtained through the homography decomposition) towards the desired pose in the 3D workspace while maintaining the visibility of (only) a virtual point located at the origin of the target object. The virtual point is used to control two degrees of rotation of the camera (around $x-$ and $y-$axes) and the third rotation axis (around camera optical axis) is controlled using the rotation matrix retrieved from homography. This technique yields a straight line trajectory for the virtual point and, hence, keeps the virtual point always in the camera's field of view. However, the camera can get too close to the target so that some features may get lost. Switching between visual servoing strategies or using repulsive potentials can be employed to avoid such situations, however, without ensuring straight line trajectories.

In [1] a similar approach has been proposed based on homography decomposition in which helicoidal shape paths (instead of straight path) are chosen as the reference path to represent camera translation from the initial position to the desired position. One should note that since the homography is known only up to an *unknown* scale, the actual camera path is not completely known and one can only determine its shape. However, regardless of the value of unknown scale factor, the entire image path will remain the same and since the control is defined directly in the image, the positioning task can be successfully accomplished given a feasible image path. In [4] a particular decomposition of homography is used to interpolate a path for a planar object with known model from the initial image to the desired final image. Given the known object model, the interpolated desired path is then transformed to a camera path by using 3D reconstruction. The camera path can then be checked for workspace boundary singularities.

Projective invariance. Malis [45] proposed an image-based path-planning approach in an invariant space defined through a projective transformation. The basic idea of using projective invariance is to create a task function which is invariant to camera intrinsic parameters and only depends on the position of the camera with respect to the observed object and on its 3D structure. This allows one to generate a path for a feature vector in the invariant space (independent of camera's intrinsic parameters) which, when followed, results in a straight line path for the camera in the workspace. The visibility of the features is (partially) achieved using a motorized zooming mechanism available on the vision system.

The main advantage of direct path-planning in image space is the independence of such approaches from camera calibration and/or object model. On the other hand, since the planning is done directly in the image space, robot/physical constraints cannot be handled through such approaches and these techniques are shown to be ineffective in complex visual servoing scenarios.

11.3.2 Optimization-based Path-planning

Planning optimal paths has absorbed a great amount of interest in robotics community. In a visual servoing task, there might be many different paths, which when followed, will result in successful accomplishment of the same task. This motivates optimization techniques aimed at finding the optimal path with respect to various costs such as distance from the image boundary, length of the path traversed by the robot, energy expenditure, *etc*.

In an early work [60], a path-planning framework is proposed based on the concept of perceptual control manifold (PCM) defined on the product of the robot's joint space and the space of all image features related to a target object. PCM can be considered as a mapping which relates a robot configuration to the vector of image features visible at that configuration. Given the model of the camera, the object, and the robot kinematic model, the PCM needs to be computed only once (in an eye-to-hand configuration) and is then applicable to any manipulation task. Constraints such as the camera's field of view and the robot joint limits and/or physical obstacles are mapped into the PCM to yield a subset of PCM as the feasible solution space. This mapping could be quite time consuming considering the number of constraints and the robot's degrees of freedom. Various optimization criteria such as minimum velocity, minimum interception time, and minimum robot movement have been considered to plan optimal paths in the feasible subset of the PCM. The proposed approach has been considered for the task of intercepting a moving target (with a known trajectory) using the visual feedbacks obtained from a fixed camera which simultaneously views both the robot's end-effector and the moving target.

In [50] closed-form collineation paths corresponding to minimum energy and minimum acceleration camera paths are planned in the image space. The proposed strategy is then generalized to the case where a number of relay (intermediate) images are available in addition of the initial and desired images. The proposed

approach guarantees convergence, however, it does not take visibility constraints into account and image features might leave the camera's field of view.

In [63] a motion generation approach called visual motion planning has been proposed to plan optimal image paths for mobile robots under motion and visibility constraints. The constraints on the motion of the robot along with the field of view limits are described in form of a number of equalities and inequalities. An optimization problem is then solved numerically using Lagrange Multipliers to obtain optimal image paths minimizing a given weighted sum cost function (here kinetic energy). The proposed approach has been applied only to mobile robots moving in 2D and 3D environments.

To pose the problem of path-planning for visual servoing as an optimization problem some researchers have introduced various parameterizations of camera trajectories. A polynomial parametrization of the scaled camera paths has been proposed in [14] where the translational path is linearly interpolated and Cayley's rotation representation is employed to rationally parameterize the rotation paths. This allows the distance of the image trajectories from the boundary of image for a single path to be easily calculated as the root of some polynomials. Hence, an optimization problem is then formulated to maximize the distance to the boundary of the image with respect to all parameterized paths. By following the planned image path, the camera follows a straight line in the workspace in the absence of calibration errors. In presence of calibration errors, the camera does not follow a straight line but moves along a different curve whose distance from the planned line grows as the calibration errors increase.

In [12] an optimal path-planning approach is proposed which allows one to consider constraints on the camera's field of view, workspace and joint limits, in the form of inequalities, together with the objective of minimizing trajectory costs including spanned image area, trajectory length, and curvature. A polynomial parametrization is devised to represent all the camera paths connecting the initial and desired locations (up to a scale factor) through an object reconstruction from image measurements and, if available, the target model. Occlusion constraints and collision avoidance for the whole robot's body cannot be represented (in the form of inequality constraints) in their formulation. Moreover, the devised optimization is nonconvex which may lead to multiple feasible regions and multiple locally optimal solutions within each region and, hence, it makes it very difficult to find the global optimal solution across all feasible regions.

In a similar work [10], a general parameterizations of trajectories from the initial to the desired location is proposed via homogeneous forms and a parameter-dependent version of the Rodrigues formula. The constraints are modeled using positivity conditions on suitable homogeneous forms. The solution trajectory is obtained by solving a linear matrix inequality (LMI) test which is a convex optimization. The proposed approach allows one to maximize some desired performances such as distance of features from the boundary of the image, camera's distance from obstacles, and similarity between the planned trajectory and a straight line.

Ideas from optimal control theory have been employed to devise image trajectories for visual servoing under visibility constraints. Planning shortest path for a differential drive robot (DDR) maintaining the visibility of a landmark using a camera with limited field of view has been considered in [3]. It is shown that the set of shortest (optimal) paths for this system consist of curve segments that are either straight-line segments or that saturate the camera's field of view. The latter correspond to exponential spirals known as T-curves. In [42] these shortest paths are followed using a switched homography-based visual servo controller. The controls that move the robot along these paths are devised based on the convergence of the elements of the homography matrix relating the current image to the final desired image. In a recent work [28], a complete motion planner for a DDR is proposed in which optimal curve segments obtained in [3] are used as motion primitives to devise locally optimal paths in an environment cluttered with obstacles. The necessary and sufficient conditions for the feasibility of a path for the DDR in the presence of obstacles and with visibility constraints (*i.e.*, sensing range and field of view limits) are also provided. In their proposed planner, occlusions due to workspace obstacles are not considered and the obstacles are assumed to be transparent.

In [56] the set of optimal curves obtained in [3] are extended and also described in the image space, so as to enable their execution using an IBVS controller directly in the image space. Feedback control along these optimal paths in the image is achieved through a set of Lyapunov controllers, each of which is in charge of a specific kind of maneuver. Nonetheless, the complete characterization of all the shortest paths and their analytic descriptions remain unsolved for a DDR.

Although the above optimization-based path-planning techniques provide a better insight into the complexity of the problem and feasible optimal paths, they are more or less limited to simple scenarios and systems. Introducing general robot/physical constraints greatly adds to the complexity of the optimization problem and, hence, accounting for such constraints in the above frameworks would greatly increase the time complexity of such techniques.

11.3.3 Potential Field-based Path-planning

In the field of robot path-planning, potential field method has been proposed as a promising local and fast obstacle avoidance strategy to plan safe and real-time motions for a robot in a constrained environment [33]. The main idea is to construct an artificial potential field defined as the sum of attractive potentials, pulling the robot towards the desired location, and repulsive potentials, pushing the robot away from various constraints such as the obstacles or robot's joint limits. A driving force computed along the negated gradient of the potential field moves the robot towards the goal location.

Mezouar and Chaumette [49] introduced robust image-based control based on the Potential Field method for a robotic arm with eye-in-hand configuration. In their proposed approach, two types of constraints are considered: field of view and robot's joint limits. To obtain valid robot trajectories, the motion of the robot is first planned

in the workspace and then projected into the image space. The attractive potentials are defined in the workspace to pull the robot towards the final desired configuration. To account for field of view limits, repulsive potentials are defined in the image space pushing the image trajectories away from the image boundary. Joint limits are avoided by imposing repulsive potentials in the joint space of the robot. So, the total force applied to the robot is a weighted sum of the individual forces computed as the negated gradient of the above potentials. The image trajectories are obtained in an iterative scheme by moving along the direction of the total force applied to the robot. The discrete image trajectories are then time scaled and tracked using an IBVS technique. The above strategy has been applied to targets with known as well as unknown models. In the latter case, a scaled Euclidean reconstruction is employed to obtain scaled camera paths in the workspace. Image local minima are automatically avoided by updating the image Jacobian using the values of the current desired image features along the time scaled feature trajectories.

As an inherent deficiency of potential field-based path-planning method, the above strategy might lead to local minima. Although the authors reported no encounter of such local minima in their experiments, imposing physical constraints such as collisions with obstacles and occlusions highly increase the chance of having local minima in the overall potential field.

In [19] a potential field-based strategy is employed to account for workspace obstacles, field of view limits, and robot's joint limits in a global planning framework. To escape local minima generated by addition of the attractive and repulsive forces, *simulated annealing* [34] is employed in which proper tuning of the initial temperature and the cooling rates are required to probabilistically ensure the method to escape from local minima and converge to the global minimum. In the proposed planning framework two different trajectory generation strategies are employed: method A, where a trajectory for the end-effector is planned with respect to the stationary target frame, and method B, where a trajectory for the target is planned with respect to the current end-effector frame. The former results in a camera path close to a straight line in the workspace, while in the latter the image trajectory of the target's origin is constrained to move as close as possible to a straight line in the image which lessens the chance of image features leaving the camera's field of view. A local switching strategy is devised to switch from image-based control to position-based control when closeness to image local minima and image singularities are detected along the planned trajectories. This is done only once to avoid instability due to repetitive switching, however there is no complete guarantee that the field of view and joint limits are always ensured after the system is switched to position-based control.

One of the main advantages of potential field-based approaches is the fast computation of driving force which makes these approaches suitable for real-time applications such as visual servoing. For example, the above strategy can be employed when tracking image trajectories to account for possible deviations from the planned trajectory due to uncertainties in modeling and/or calibration (*e.g.* [12]).

11.3.4 Global Path-planning

The convergence problems and deficiencies of the above path-planning techniques in accounting for all the constraints in visual servoing tasks motivates the need for general and global path-planning approaches. A great deal of research has been carried out on global path-planning for various robotic systems within the path-planning community, see *e.g.* [39] and [40]. Here we report on some of these techniques which have been successfully incorporated into the visual servoing framework.

A global stabilizing strategy using navigation functions is presented in [17] which guarantees convergence to a visible goal from almost every initial visible configuration while maintaining viability of all features along the way without following a predefined reference image trajectory. One should note that constructing such navigation functions is limited to very simple scenarios only.

In [2] a probabilistic roadmap approach has been utilized to plan minimal-occlusion paths for an in-hand camera with respect to a target object. They employ the technique proposed in [61] to compute the boundary separating the visible regions (from where the target is visible) from the occluded regions (from where the target is not visible due to occlusion by workspace obstacles). Their proposed algorithm then assigns penalties to camera's trajectories within a given probabilistic roadmap (for camera translation) proportional to the distance the camera travels while outside the visible region. One should note that camera's orientation or field of view limits are not taken into account in their proposed approach.

Inspired by the work in [62] on global path-planning with general end-effector constraints, we incorporated sampling-based global path-planning with visual servoing for a robotic arm equipped with an in-hand camera and proposed a planner [32] which explores the camera space for camera paths satisfying field of view limits and occlusion constraints, and utilizes a local planner to track these paths in the robot's joint space to ensure feasible motions of the robot while accounting for robot's joint limits and collision with obstacles. The result is a search tree as in [35] which alternatively explores the camera and joint spaces (see Figure 11.1). The camera path connecting the initial and desired camera poses is then extracted from the tree and is projected into the image space to obtain sampling-based feature trajectories as a sequence of image waypoints. The image space waypoints are then time parameterized and scaled using *cubic splines*. The spline feature trajectories are tracked using an IBVS technique (as in [49]) at the execution stage.

We demonstrated via simulations and real experiments [32] that the robot is able to visual servo to a desired configuration while avoiding occlusions of target, keeping the target within the camera's field of view, and avoiding collision with obstacles. Such capabilities enhances the applicability of visual servoing to significantly more complex environments/tasks. In the proposed approach, we assumed that the 3D model of the target object and the camera's intrinsic parameters are known *a priori*. The 3D model of the object is required to estimate the corresponding camera poses at the initial and desired views. Furthermore, these parameters are required

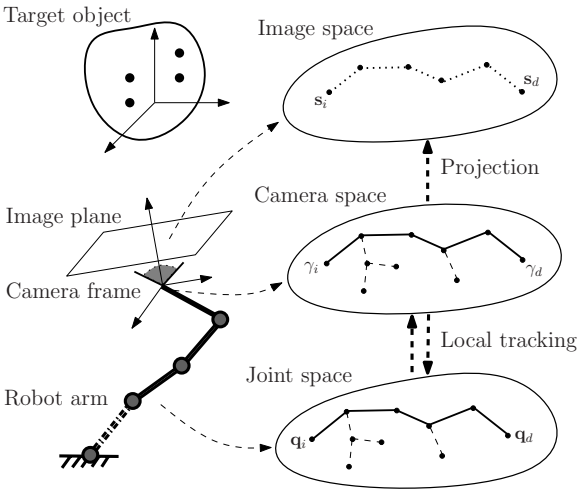


Fig. 11.1 Our proposed planner [32] alternatively explores the camera space and joint space for feasible robot/camera paths to obtain feasible feature trajectories in image space

to transform camera paths to waypoints on the image feature trajectories. We also assume that obstacles in the workspace are known *a priori*.

To be successful, global path-planning approaches require a complete and (relatively) precise knowledge of the environment, camera calibration, and object model. These requirements can be limiting in many visual servoing scenarios. The need for such exact knowledge can be relaxed by accounting for modeling and calibration uncertainties at planning stage. In the following section we discuss the effects of uncertainties in visual servoing, especially in tracking planned trajectories at the execution time, and report on a few recent works on path-planning under uncertainties for visual servoing.

11.4 Path-planning under Uncertainty for Visual Servoing

Planned paths need to be executed and the robot may not exactly follow the planned path due to the uncertainties and in fact, in some cases, the followed paths in the workspace and in the image space can be quite different from the planned ones thereby resulting in violation of some of the constraints even if they have been fulfilled at the planning stage. The influence of errors in intrinsic and extrinsic camera parameters on the performance of visual servoing scheme has been examined in [20]. In [37] the propagation of image error through pose estimation and visual servoing control law has been analyzed.

A number of researchers have proposed local and real-time techniques to account for likely deviations from the planned path at the execution (tracking) stage. For example, these could be locally taken care of using potential field type techniques, however as mentioned before, such techniques are prone to failure due to

local minima [49]. An alternative framework to deal with unmodelled uncertainties is to retreat the robot/camera and/or re-plan quickly when encountering violation in a constraint. A variable zooming technique was suggested by [38] to bring the target back within the visibility range if occluded by an obstacle. This zooming effect can also drastically improve the performance of the underlying image-based visual servoing technique by reducing the measurement noise in fixed-size objects viewed by a camera from distance.

Comport et al. [15] proposed an augmented reality approach for visual servoing. Although their approach mainly focuses on camera pose estimation by means of a virtual visual servoing method, but this can be extended to scenarios in which some feature points on the target may go out of sight temporarily, *e.g.* due to unmodelled uncertainties. An augmented reality approach can then be utilized to virtually position the missing feature points in the image based on rudimentary information obtained from other objects in a scene cluttered with known features, *i.e.* straight edges, *etc*. In this case, the target is used as the primary object for visual servoing while other image features can contribute to the pose estimation, and eventually to the servoing task, when a finite number of feature points fall off the cameras field of view.

Nonlinear model predictive control strategies have been proposed to account for uncertainties in planned trajectories in visual servo control loop as well, *e.g.* [57]. Systems' parameters would be corrected beyond a temporal receding horizon (*i.e.*, the time span during which the optimal control action is computed and executed) after each iteration. The discrepancy between the predicted system's behavior based on the computed control action and that in real implementation is then used to further correct the estimates of the system's parameters. The time required to estimate these parameters via a nonlinear optimization technique must be way shorter than the receding horizon in which this optimization is carried out. Otherwise, the applicability of this technique for real-time scenarios would be questionable. Developing a guideline for selecting the optimal size for the receding horizon for robust visual servoing in real time remains an open research area.

Robustness with respect to calibration errors in terms of the tracking error boundedness along the planned trajectories has been considered in [53]. Given a user defined bound on the tracking error, they propose a control strategy to modulate control gains and/or the desired tracking velocity to guarantee error boundedness. Through the proposed velocity modulation technique, one could use low control gains while keeping the tracking error bounded. While this technique and those mentioned above, to some extent, are expected to account for the deviations from the planned trajectories in the image space, the deviations from the physical space trajectories can cause robot/physical constraints violations. The above mentioned local strategies for accounting deviations from planned path are either prone to local minima or not general enough to account for all types of constraints (and the related uncertainties), in particular robot/physical constraints. Hence, there is need for taking the uncertainties into account in a global as well as general manner at the planning stage.

The planned paths obtained based on only a nominal model of the camera and/or robot may not be fully traversable by the robot without violating certain constraints. In a recent work [9], Chesi proposed a planning approach to design a robust image trajectory that satisfies the required constraints not only for the nominal model but rather for a family of admissible models. In the proposed approach an uncertain model has been considered for image correspondence between the initial and desired images, and the camera's intrinsic parameters are assumed to be affected by some unknown random errors with known bounds. Given the above uncertain models, there are different admissible camera poses and consequently different camera trajectories rather than a common and robust one. A polynomial parametrization is proposed through which each camera trajectory is parameterized by a possible camera pose and by a design variable which is common to all admissible trajectories. So, the robust trajectory is computed through an optimization problem determining the common design variable that satisfies field of view limits and maximizes the distance of the image features from the boundary of image on all parameterized trajectories.

Although the results obtained through the above approaches in taking calibration uncertainty and measurement errors into account seem promising, more research needs to be done. Physical constraints, especially collisions and occlusions, are highly affected by the uncertainties in the modeling of the environment. Robot path-planning considering uncertainties in modeling, localization, and sensing has been studied for decades within path-planning community [40] yielded a number of promising approaches, *e.g.* [5], [30], [43], [48], [55]. Incorporating the research results achieved through these approaches into the visual servoing framework would be a promising future direction. Moreover, planning robust trajectories for visual servoing tasks in unknown or partially known environments remains an open research problem.

11.5 Conclusions

We provided a comprehensive review of existing path-planning for visual servoing techniques aimed at making the visual servoing more robust in complex scenarios, especially in applications where the initial and desired views are distant. Considering the underlying path-planning approach, the existing techniques have been divided into four categories: (1) image space path-planning, (2) optimization-based path-planning, (3) potential field-based path-planning, and (4) global path-planning. We reported on the previous works pertinent to each category and for each technique we discussed the set of assumptions along with its benefits and drawbacks and its integration with the reactive visual servo controllers.

Recent works (discussed in Section 11.4) demonstrated the effectiveness of accounting for modeling/calibration uncertainties and measurement errors at the planning stage in generating robust trajectories for visual servoing scenarios where the available data are affected by uncertainties. Towards that aim, incorporating the

results achieved on robot path-planning under uncertainty within the path-planning community is a promising direction to follow.

References

[1] Allotta, B., Fioravanti, D.: 3D motion planning for image-based visual servoing tasks. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 2173–2178 (2005)
[2] Baumann, M., Dupuis, D., Leonard, S., Croft, E., Little, J.: Occlusion-free path planning with a probabilistic roadmap. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 2151–2156 (2008)
[3] Bhattacharya, S., Murrieta-Cid, R., Hutchinson, S.: Optimal paths for landmark-based navigation by differential-drive vehicles with field-of-view constraints. IEEE Trans. Robot. 23(1), 47–59 (2007)
[4] Borgstadt, J., Ferrier, N.: Visual servoing: path interpolation by homography decomposition. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 723–730 (2001)
[5] Censi, A., Calisi, D., Luca, A., Oriolo, G.: A Bayesian framework for optimal motion planning with uncertainty. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 1798–805 (2008)
[6] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman, D., Hager, G., Morse, A. (eds.) Confluence of Vision and Control, pp. 66–78. Springer, Heidelberg (1998)
[7] Chaumette, F., Hutchinson, S.: Visual servo control Part I: Basic approaches. IEEE Robot. Autom. Mag. 13(4), 82–90 (2006)
[8] Chaumette, F., Hutchinson, S.: Visual servo control Part II: Advanced approaches. IEEE Robot. Autom. Mag. 14(1), 109–118 (2007)
[9] Chesi, G.: Desigining image trajectoris in the presence of uncertain data for robust visual servoing path planning. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 1492–1497 (2009)
[10] Chesi, G.: Visual servoing path planning via homogeneous forms and LMI optimizations. IEEE Trans. Robot. 25(2), 281–291 (2009)
[11] Chesi, G., Hashimoto, K., Prattichizzo, D., Vicino, A.: Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Trans. Robot. 20(5), 908–914 (2004)
[12] Chesi, G., Hung, Y.: Global path-planning for constrained and optimal visual servoing. IEEE Trans. Robot. 23(5), 1050–1060 (2007)
[13] Chesi, G., Malis, E., Cipolla, R.: Automatic segmentation and matching of planar contours for visual servoing. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 2753–2758 (2000)
[14] Chesi, G., Prattichizzo, D., Vicino, A.: Straight line path-planning in visual servoing. Trans. ASME, J. Dyn. Syst. Meas. Control 129(4), 541–543 (2007)
[15] Comport, A., Marchand, E., Pressigout, M., Chaumette, F.: Real-time markerless tracking for augmented reality: the virtual visual servoing framework. IEEE Trans. Vis. Comput. Graphics 12(4), 615–628 (2006)
[16] Corke, P., Hutchinson, S.: A new partitioned approach to image-based visual servo control. IEEE Trans. Robot. Autom. 17(4), 507–515 (2001)
[17] Cowan, N., Weingarten, J., Koditschek, D.: Visual servoing via navigation functions. IEEE Trans. Robot. Autom. 18(4), 521–533 (2002)

[18] Deguchi, K.: Optimal motion control for image-based visual servoing by decoupling translation and rotation. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 705–711 (1998)
[19] Deng, L., Janabi-Sharifi, F., Wilson, W.: Hybrid motion control and planning strategies for visual servoing. IEEE Trans. Ind. Electron. 52(4), 1024–1040 (2005)
[20] Espiau, B.: Effect of camera calibration errors on visual servoing in robotics. In: Proc. 3rd Int. Symp. Exp. Robot., pp. 182–192 (1994)
[21] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Trans. Robot. Autom. 8(3), 313–326 (1992)
[22] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Trans. Robot. Autom. 8(3), 313–326 (1992)
[23] Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press, Cambridge (1993)
[24] Gans, N., Hutchinson, S.: Stable visual servoing through hybrid switched-system control. IEEE Trans. Robot. 23(3), 530–540 (2007)
[25] Gans, N., Hutchinson, S., Corke, P.: Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control. Int. J. Robot. Res. 22(10-11), 955–981 (2003)
[26] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, Cambridge (2003)
[27] Hashimoto, K., Noritsugu, T.: Potential problems and switching control for visual servoing. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 423–428 (2000)
[28] Hayet, J., Esteves, C., Murrieta-Cid, R.: A motion planner for maintaining landmark visibility with a differential drive robot. In: To be published in Algorithmic Foundations of Robotics VIII (2009)
[29] Hosoda, K., Sakamoto, K., Asada, M.: Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 29–34 (1995)
[30] Huang, Y., Gupta, K.: RRT-SLAM for motion planning with motion and map uncertainty for robot exploration. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 1077–1082 (2008)
[31] Hutchinson, S., Hager, G., Corke, P.: A tutorial on visual servo control. IEEE Trans. Robot. Autom. 12(5), 651–670 (1996)
[32] Kazemi, M., Gupta, K., Mehrandezh, M.: Global path planning for robust visual servoing in complex environments. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 326–332 (2009)
[33] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. Int. J. Robot. Res. 5(1), 90–98 (1986)
[34] Kirkpatrick, S., Gelatt Jr., C., Vecchi, M.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)
[35] Kuffner, J., LaValle, S.: RRT-connect: an efficient approach to single-query path planning. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 995–1001 (2000)
[36] Kyrki, V., Kragic, D., Christensen, H.: New shortest-path approaches to visual servoing. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 349–354 (2004)
[37] Kyrki, V., Kragic, D., Christensen, H.: Measurement errors in visual servoing. Robot. Auton. Syst. 54(10), 815–827 (2006)
[38] Lang, O., Graser, A.: Visual control of 6 DOF robots with constant object size in the image by means of zoom camera. In: Proc. Conf. IEEE Ind. Electron. Society, pp. 1342–1347 (1999)

[39] Latombe, J.C.: Robot Motion Planning. Kluwer Academic Publishers, Norwell (1991)
[40] LaValle, S.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
[41] LaValle, S., Gonzalez-Banos, H., Becker, C., Latombe, J.C.: Motion strategies for maintaining visibility of a moving target. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 731–736 (1997)
[42] Lopez-Nicolas, G., Bhattacharya, S., Guerrero, J., Sagues, C., Hutchinson, S.: Switched homography-based visual control of differential drive vehicles with field-of-view constraints. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 4238–4244 (2007)
[43] Lozano-Perez, T., Mason, M., Taylor, R.: Automatic synthesis of fine-motion strategies for robots. Int. J. Robot. Res. 3(1), 3–24 (1984)
[44] Mahony, R., Hamel, T., Chaumette, F.: A decoupled image space approach to visual servo control of a robotic manipulator. In: Proc. IEEE Int. Conf. Robot. Autom., pp. 3781–3786 (2002)
[45] Malis, E.: Visual servoing invariant to changes in camera-intrinsic parameters. IEEE Trans. Robot. Autom. 20(1), 72–81 (2004)
[46] Malis, E., Chaumette, F.: 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. Int. J. Comput. Vis. 37(1), 79–97 (2000)
[47] Malis, E., Chaumette, F., Boudet, S.: 2-1/2-D Visual servoing. IEEE Trans. Robot. Autom. 15(2), 238–250 (1999)
[48] Melchior, N., Simmons, R.: Particle RRT for path planning with unceratinty. Proc. IEEE Int. Conf. Robot. Autom., 1617–1624 (2007)
[49] Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Trans. Robot. Autom. 18(4), 534–549 (2002)
[50] Mezouar, Y., Chaumette, F.: Optimal camera trajectory with image-based control. Int. J. Robot. Res. 22(10-11), 781–803 (2003)
[51] Michel, P., Scheurer, C., Kuffner, J., Vahrenkamp, N., Dillmann, R.: Planning for robust execution of humanoid motions using future perceptive capability. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 3223–3228 (2007)
[52] Morel, G., Liebezeit, T., Szewczyk, J., Boudet, S., Pot, J.: Explicit incorporation of 2D constraints in vision based control of robot manipulators. In: Proc. 6th Int. Symp. Exp. Robot., pp. 99–108 (2000)
[53] Morel, G., Zanne, P., Plestan, F.: Robust visual servoing: bounding the task function tracking errors. IEEE Trans. Control Syst. Technol. 13(6), 998–1009 (2005)
[54] Park, J., Chung, M.: Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy. IEEE Trans. Robot. Autom. 19(2), 250–258 (2003)
[55] Pepy, R., Lambert, A.: Safe path planning in an uncertain-configuration space using RRT. In: Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 5376–5381 (2006)
[56] Salaris, P., Belo, F., Fontanelli, D., Greco, L., Bicchi, A.: Optimal paths in a constrained image plane for purely image-based parking. In: Proc. IEEE/RSJ Int. Conf. on Intell. Robot. Syst., pp. 1673–1680 (2008)
[57] Sauvee, M., Poignet, P., Dombre, E., Courtial, E.: Image based visual servoing through nonlinear model predictive control. In: Proc. IEEE Conf. Decis. Control, pp. 1776–1781 (2006)
[58] Schramm, F., Morel, G.: Ensuring visibility in calibration-free path planning for image-based visual servoing. IEEE Trans. Robot. 22(4), 848–854 (2006)
[59] Sharma, R., Hutchinson, S.: Motion perceptibility and its application to active vision-based servo control. IEEE Trans. Robot. Autom. 13(4), 607–617 (1997)

[60] Sharma, R., Sutanto, H.: A framework for robot motion planning with sensor constraints. IEEE Trans. Robot. Autom. 13(1), 61–73 (1997)
[61] Tarabanis, K., Tsai, R., Kaul, A.: Computing occlusion-free viewpoints. IEEE Trans. Pattern Anal. Mach. Intell. 18(3), 279–292 (1996)
[62] Yao, Z., Gupta, K.: Path planning with general end-effector constarints. Robot. Auton. Syst. 55(4), 316–327 (2007)
[63] Zhang, H., Ostrowski, J.: Visual motion planning for mobile robots. IEEE Trans. Robot. Autom. 18(2), 199–208 (2002)

Chapter 12
Single Camera Structure and Motion Estimation

Ashwin P. Dani and Warren E. Dixon

Abstract. Two new continuous nonlinear observers are proposed for the problem of structure from motion (SfM) and structure and motion (SaM) of a stationary object observed by a moving camera. The observer for SfM, where full velocity feedback is available, yields global exponential convergence of the states for the structure. The SaM observer requires only one of the linear velocities as a feedback and identifies the states asymptotically. The linear velocity is used to derive the scene scale information. The observer gain conditions are derived to prove the stability of the proposed observers through a Lyapunov-based analysis.

12.1 Introduction

Provided points of interest can be tracked between multiple images (*i.e.*, through feature point tracking[1]), image feedback can be used to estimate the motion of the tracked feature points or can be used to estimate the geometry of the feature points. The objective of the *SfM* problem is to estimate the Euclidean geometry (*i.e.*, 3D structure) of the tracked feature points attached to an object provided the relative motion[2] between the camera and the object is known. The converse of the SfM problem is the *motion from structure (MfS)* problem where the relative motion between the camera and the object is estimated based on known geometry of the tracked

Ashwin P. Dani and Warren E. Dixon
Department of Mechanical and Aerospace Engineering, University of Florida,
Gainesville, FL 32608, USA
e-mail: {ashwin31,wdixon}@ufl.edu

[1] Some well-known existing solutions to the feature point tracking problem include the Kanade–Lucas–Tomasi (KLT) tracker [43], the scale invariant feature transformation (SIFT) tracker [27], and the speed up robust features (SURF) tracker [4].

[2] Throughout this chapter, relative motion is between a moving camera and stationary object (*i.e.*, the camera-in-hand problem) or a stationary camera and moving object (*i.e.*, the camera-to-hand problem).

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 209–229.
springerlink.com

feature points attached to an object. An extended problem is *SaM* where the objective is to estimate the Euclidean geometry of the tracked feature points as well as the relative motion between the camera and tracked feature points. The SaM problem is a fundamental problem and some examples indicate that SaM estimation is only possible up to a scale when a pinhole camera model is used. In this chapter, a new method to solve the SfM problem is presented where global exponential estimation is achieved with a continuous observer. The new SfM algorithm is extended to solve the partial SaM problem where the objective is to asymptotically identify the Euclidean geometry of the tracked feature points and the camera motion, provided at least one linear velocity of the camera is known.

Solutions to the SfM, MfS, and the SaM problems can be broadly classified as offline methods (batch methods) and online methods (iterative methods). References and critiques of batch methods can be found in [16, 17, 32, 21, 31, 42, 2, 3] and the references therein. Batch methods extract an image data set from a given image sequence and then the 3D structure is reconstructed from the data set. These methods are usually based on nonlinear optimization, projective methods, or invariant-based methods. Often batch methods lack an analytical analysis of convergence, with the exception of results such as [37, 22, 33] using convex optimization techniques. The main drawback of batch methods is that they cannot be used to execute online/real-time tasks. Thus, the need arises for iterative or online methods with analytical guarantees of convergence.

Online methods typically formulate the SfM and MfS problems as continuous time differential equation, where the image dynamics are derived from a continuous image sequence (see [39, 18, 8, 40, 1, 6, 12, 34, 10, 11, 20] and the references therein). Online methods often rely on the use of an extended Kalman filter (EKF) [39, 30, 23, 9]. Kalman filter based approaches also lack a convergence guarantee and could converge to wrong solutions in practical scenarios. Also, *a priori* knowledge about the noise is required for such solutions. In comparison to Kalman filter-based approaches, some researchers have developed nonlinear observers for SfM with analytical proofs of stability. For example, a discontinuous high-gain observer called identifier-based observer (IBO) is presented for structure estimation in [20] under the assumption of known camera motion. In [6], a discontinuous sliding-mode observer is developed which guarantees exponential convergence of the states to an arbitrarily small neighborhood, *i.e.*, uniformly ultimately bounded (UUB) result. A continuous observer which guarantees asymptotic structure estimation is presented in [12] under the assumption of known camera motion. The observer in [12] was modified in [18] to asymptotically identify the structure of a target given the relative motion of an omnidirectional (paracadioptric) camera. An asymptotically stable reduced-order observer is presented in [24] to estimate the structure given known camera motion. Under the assumption that a known Euclidean distance between two feature points is known, a nonlinear observer is used in [8] to asymptotically identify the camera motion.

Various batch and iterative methods have been developed to solve the SaM problem up to a scale, such as [42, 2, 3, 34, 35]. However, in comparison to SfM and MfS results, sparse literature is available where the SaM problem is formulated in

terms of continuous image dynamics with associated analytical stability analysis. Recently, a nonlinear observer was developed in [10] to asymptotically identify the structure given the camera motion (*i.e.*, the SfM problem) or to asymptotically identify the structure and the unknown time-varying angular velocities given the linear velocities. In another recent result in [28], an IBO approach [20] was used to estimate the structure and the constant angular velocity of the camera given the linear velocities.

Two continuous nonlinear observers are developed in this chapter. The first observer estimates the structure of the object provided the linear and angular camera velocities are known with respect to a stationary object (*i.e.*, SfM). A Lyapunov-based analysis is provided that illustrates global exponential stability of the observer errors provided some observability conditions are satisfied. This observer is extended to address the SaM problem where the structure, the time-varying angular velocities and two of the three unknown time-varying linear velocities are estimated (*i.e.*, one relative linear velocity is assumed to be known). A Lyapunov-based analysis is provided that indicates the SaM observer errors are globally asymptotically regulated provided some additional (more restrictive) observability conditions are satisfied.

The chapter is organized in the following manner. In Section 12.2 relationships are developed between the 3D Euclidean coordinates and image space coordinates. Section 12.3 describes the perspective camera motion model. Section 12.4 states the objective of SfM and SaM, followed by Section 12.4.1 and Section 12.4.2 which propose continuous nonlinear observers for SfM and SaM problems, respectively, and the associated stability analyses.

12.2 Euclidean and Image Space Relationships

The development in this chapter is based on the scenario depicted in Figure 12.1 where a moving camera views four or more planar and noncollinear feature points (denoted by $j = \{1, 2, ..., n\}\ \forall n \geq 4$) lying fixed in a visible plane π_r attached to an object in front of the camera. In Figure 12.1, $\mathcal{F}_r$ is a static coordinate frame attached to the object. A static reference orthogonal coordinate frame $\mathcal{F}_c^*$ is attached to the camera at the location corresponding to an initial point in time t_0 where the object is in the camera field of view (FOV). After the initial time, an orthogonal coordinate frame $\mathcal{F}_c$ attached to the camera[3] undergoes some rotation $\bar{R}(t) \in SO(3)$ and translation $\bar{x}_f(t) \in \mathbb{R}^3$ away from $\mathcal{F}_c^*$. The rotation between the camera frame $\mathcal{F}_c$ and the object frame $\mathcal{F}_r$ is denoted by $R(t) \in SO(3)$ and the constant rotation between the camera frame $\mathcal{F}_c^*$ and the object frame $\mathcal{F}_r$ is denoted by $R^* \in SO(3)$. Likewise, the translation vector between $\mathcal{F}_c$ and $\mathcal{F}_r$ is given as $x_f(t) \in \mathbb{R}^3$, and the constant translation between $\mathcal{F}_c^*$ and $\mathcal{F}_r$ is given as $x_f^* \in \mathbb{R}^3$. The constant normal vector to plane π_r, measured in $\mathcal{F}_c^*$, is given by $n^* \in \mathbb{R}^3$, where $d^* \in \mathbb{R}^3$ denotes the constant distance between $\mathcal{F}_c^*$ and π_r along the normal.

[3] $\mathcal{F}_c^*$ and $\mathcal{F}_c$ are collocated at t_0.

The constant Euclidean coordinates $\bar{m}_j^* \in \mathbb{R}^3$ of the feature points expressed in the camera frame $\mathcal{F}_c^*$ are defined as

$$\bar{m}_j^* = \left[x_{1j}^*, x_{2j}^*, x_{3j}^* \right]^T.$$

These feature points, projected on the image plane π_i, are given by the constant normalized coordinates $m_j^* \in \mathbb{R}^3$ as

$$m_j^* = \left[\frac{x_{1j}^*}{x_{3j}^*}, \frac{x_{2j}^*}{x_{3j}^*}, 1 \right]^T. \tag{12.1}$$

The Euclidean coordinates $\bar{m}_j(t) \in \mathbb{R}^3$ of the feature points expressed in the camera frame $\mathcal{F}_c$ and the respective normalized Euclidean coordinates $m_j(t) \in \mathbb{R}^3$ are defined as

$$\bar{m}_j(t) = \left[x_{1j}(t), x_{2j}(t), x_{3j}(t) \right]^T, \tag{12.2}$$

$$m_j(t) = \left[\frac{x_{1j}(t)}{x_{3j}(t)}, \frac{x_{2j}(t)}{x_{3j}(t)}, 1 \right]^T. \tag{12.3}$$

Consider a closed and bounded set $\mathcal{Y} \subset \mathbb{R}^3$. To facilitate the subsequent development, auxiliary state vectors $y_j^* = [y_{1j}^*, y_{2j}^*, y_{3j}^*]^T \in \mathcal{Y}$ and $y_j(t) = [y_{1j}(t), y_{2j}(t), y_{3j}(t)]^T \in \mathcal{Y}$ are constructed from (12.1) and (12.3) as

$$y_j^* = \left[\frac{x_{1j}^*}{x_{3j}^*}, \frac{x_{2j}^*}{x_{3j}^*}, \frac{1}{x_{3j}^*} \right]^T \qquad y_j = \left[\frac{x_{1j}}{x_{3j}}, \frac{x_{2j}}{x_{3j}}, \frac{1}{x_{3j}} \right]^T. \tag{12.4}$$

The corresponding feature points m_j^* and $m_j(t)$ viewed by the camera from two different locations (and two different instances in time) are related by a depth ratio $\alpha_j(t) \in \mathbb{R}$ and a homography matrix $H(t) \in \mathbb{R}^{3\times 3}$ as

$$m_j = \underbrace{\frac{x_{3j}^*}{x_{3j}}}_{\alpha_j(t)} \underbrace{\left(\bar{R} + \frac{\bar{x}_f}{d^*} n^{*T} \right)}_{H} m_j^*. \tag{12.5}$$

Using projective geometry, the normalized Euclidean coordinates m_j^* and $m_j(t)$ can be related to the pixel coordinates in the image space as

$$p_j = A m_j, \qquad p_j^* = A m_j^* \tag{12.6}$$

where $p_j(t) = \left[u_j \; v_j \; 1 \right]^T$ is a vector of the image-space feature point coordinates $u_j(t), v_j(t) \in \mathbb{R}$ defined on the closed and bounded set $\mathcal{I} \subset \mathbb{R}^3$, and $A \in \mathbb{R}^{3\times 3}$ is a constant, known, invertible intrinsic camera calibration matrix [29]. Since A is

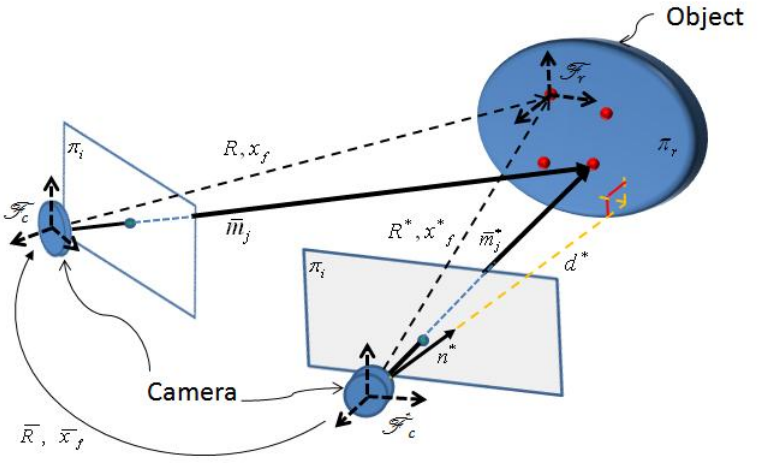


Fig. 12.1 Projective relationships of images formed at different times by moving object

known, four noncollinear feature points in a plane[4] can be used along with the relationships in (12.5) and (12.6) to develop a set of linear equations in terms of the image feedback that can be solved to compute and decompose $H(t)$. Specifically, the homography matrix can be decomposed to recover the rotation $\bar{R}(t)$ between $\mathcal{F}_c^*$ and $\mathcal{F}_c$, the normal vector n^*, the scaled translation $\frac{\bar{x}_f(t)}{d^*}$ (although the individual components $\bar{x}_f(t) \in \mathbb{R}^3$ and d^* are generally unknown), and depth ratio $\alpha_j(t)$ using standard techniques [13, 44]. The expression in (12.6) can be used to recover $m_j(t)$, which can be used to partially reconstruct the state $y(t)$ so that the first two components of $y(t)$ can be determined.

Assumption 12.1. The relative Euclidean distance $x_{3j}(t)$ between the camera and the feature points observed on the target is upper and lower bounded by some known positive constants (*i.e.*, the object remains within some finite distance away from the camera). Therefore, the definition in (12.4) can be used to assume that

$$\bar{y}_3 \geq \left| y_{3j}(t) \right| \geq \underline{y_3} \tag{12.7}$$

where $\bar{y}_3, \underline{y_3} \in \mathbb{R}$ denote known positive bounding constants. Let us define a convex hypercube Ω in $\mathbb{R}$, as

$$\Omega = \{ y_3 \mid \underline{y_3} \leq y_3 \leq \bar{y}_3 \}.$$

Likewise, since the image coordinates are constrained (*i.e.*, the target remains in the camera field of view) the relationships in (12.3), (12.4), and (12.6) along with the fact that A is invertible can be used to conclude that

[4] The homography can also be computed with 8 noncoplanar and noncollinear feature points using the "virtual parallax" algorithm.

$$\bar{y}_1 \geq \left|y_{1j}(t)\right| \geq \underline{y_1} \qquad \bar{y}_2 \geq \left|y_{2j}(t)\right| \geq \underline{y_2}$$

where $\bar{y}_1, \bar{y}_2, \underline{y_1}, \underline{y_2} \in \mathbb{R}$ denote known positive bounding constants.

Assumption 12.2. The motion of the camera is assumed to be smooth such that the velocities are assumed to be bounded by a constant. Thus, $y_j(t)$ belongs to class C^1, which also implies that the first derivatives of $y_j(t)$ is bounded by a constant. For the remainder of this chapter, the feature point subscript j is omitted to streamline the notation.

Assumption 12.3. The subsequent development is based on the assumption that both $p(t)$ and its time derivative $\dot{p}(t)$ are available, where $\dot{p}(t)$ (*i.e.*, the optic flow) is available through numerical differentiation of the image coordinates. Assuming that $p(t)$ and $\dot{p}(t)$ are available, then (12.3), (12.4), and (12.6) can be used to conclude that $y_1(t), y_2(t), \dot{y}_1(t), \dot{y}_2(t)$ can be computed.

12.3 Perspective Camera Motion Model

At some spatiotemporal instant, the camera views a point q on the object. As seen from Figure 12.2, the point q can be expressed in the coordinate system $\mathcal{F}_c$ as

$$\bar{m} = x_f + Rx_{o'q} \tag{12.8}$$

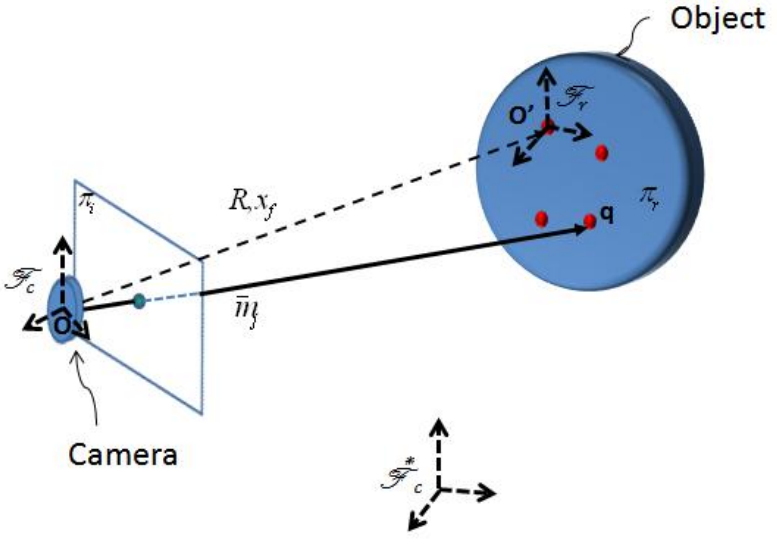


Fig. 12.2 Coordinate relationship describing the position of a point on object as seen from the camera

where $x_{o'q}$ is a vector from the origin of coordinate system $\mathscr{F}_r$ to the point q expressed in the coordinate system $\mathscr{F}_r$. Differentiating (12.8), the relative motion of q as observed in the camera coordinate system can be expressed by the following kinematics [29, 19]

$$\dot{\bar{m}} = [\omega]_\times \bar{m} + b \tag{12.9}$$

where $\bar{m}(t)$ is defined in (12.2), $[\omega]_\times \in \mathbb{R}^{3\times 3}$ denotes a skew symmetric matrix formed from the angular velocity vector of the camera $\omega(t) = \begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \in \mathscr{W}$, and $b(t) = \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix}^T \in \mathscr{B}$ denotes the linear velocity of the camera. The sets $\mathscr{W}$ and $\mathscr{B}$ are closed and bounded sets such that $\mathscr{W} \subset \mathbb{R}^3$ and $\mathscr{B} \subset \mathbb{R}^3$. By re-arranging the expression (12.9), the motion of a stationary point as observed by a moving camera can also be expressed as

$$\dot{\bar{m}} = \begin{bmatrix} 1 & 0 & 0 & 0 & x_3 & -x_2 \\ 0 & 1 & 0 & -x_3 & 0 & x_1 \\ 0 & 0 & 1 & x_2 & -x_1 & 0 \end{bmatrix} \begin{bmatrix} b \\ \omega \end{bmatrix}. \tag{12.10}$$

Using (12.4) and (12.10), the dynamics of the partially measurable state $y(t)$ can be expressed as

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{bmatrix} = \begin{bmatrix} y_3 & 0 & -y_1 y_3 & -y_1 y_2 & 1+y_1^2 & -y_2 \\ 0 & y_3 & -y_2 y_3 & -(1+y_2^2) & y_1 y_2 & y_1 \\ 0 & 0 & y_3^2 & y_2 y_3 & -y_1 y_3 & 0 \end{bmatrix} \begin{bmatrix} b \\ \omega \end{bmatrix} \tag{12.11}$$

where the states $y_1(t)$ and $y_2(t)$ can be measured as the output of the system through the transformation given by (12.6).

12.4 Structure and Motion Estimation

The SfM problem [22, 40, 1, 12, 34, 20, 30, 24, 29, 38] is defined as the desire to recover the stationary structure (*i.e.*, Euclidean coordinates $\bar{m}(t)$) from known information about the velocities $b(t)$ and $\omega(t)$ of the moving camera. Conversely, the dual problem of MfS [39, 8, 14, 15] focuses on determining the relative velocities $b(t)$ and $\omega(t)$ given some information about structure of the object, *e.g.* length between two points. It is well known that recovering the structure $\bar{m}(t)$, as well as the motion parameters $\omega(t)$ and $b(t)$, given only the series of 2D images is possible only up to a scale [29]. The projective transformation looses the scale information, and it cannot be recovered unless provided through other means, *e.g.* external calibration of the camera [36]. In fact, various scenarios can easily be constructed as examples where it is not possible to solve the SaM problem due to the lack of scale information. In this chapter, observers are designed for the perspective system given by (12.9) for two scenarios: 1) all six motion parameters are known (*i.e.*, the classic SfM problem), and 2) only a single linear velocity parameter is known (*i.e.*, SaM).

12.4.1 Estimation with Known Angular and Linear Velocities

In this section, a nonlinear SfM estimator is presented for the perspective dynamic system given by (12.11) assuming that all six velocities are known along with the intrinsic camera calibration matrix A. Scenarios where the relative motion $\omega(t)$ and $b(t)$ are known include a camera attached to the end-effector of a robot or attached to some vehicle with known motion (acquired from global positioning system (GPS), inertial measurement unit (IMU), or other sensor data).

12.4.1.1 Observer Design

The Euclidean structure $\bar{m}(t)$ can be estimated once $y_3(t)$ in (12.4) is determined, since $y_1(t)$ and $y_2(t)$ are measurable from (12.3) and (12.6). Since $y_3(t)$ appears in the image dynamics for $y_1(t)$, $y_2(t)$, and $y_3(t)$ in (12.11), the subsequent development is based on the strategy of constructing the estimates $\hat{y}(t) \in \mathbb{R}^3 = \left[\hat{y}_1(t)\ \hat{y}_2(t)\ \hat{y}_3(t)\right]^T$. To quantify this objective, an estimate error $\bar{e}(t) \in \mathbb{R}^3 \triangleq \left[e_1(t)\ e_2(t)\ e_3(t)\right]^T$ is defined as

$$e_1 \triangleq y_1 - \hat{y}_1, \quad e_2 \triangleq y_2 - \hat{y}_2, \quad e_3 \triangleq y_3 - \hat{y}_3. \tag{12.12}$$

Based on Assumption 12.3, $e_1(t)$ and $e_2(t)$ can be determined and used in the estimate design. The estimates $\hat{y}_1(t)$ and $\hat{y}_2(t)$ are generated according to the update law

$$\dot{\hat{y}}_1 = \hat{y}_3 b_1 - y_1 \hat{y}_3 b_3 - y_1 y_2 \omega_1 + (1 + y_1^2)\omega_2 - y_2 \omega_3 + k_1 e_1, \tag{12.13}$$

$$\dot{\hat{y}}_2 = \hat{y}_3 b_2 - y_2 \hat{y}_3 b_3 - (1 + y_2^2)\omega_1 + y_1 y_2 \omega_2 + y_1 \omega_3 + k_2 e_2, \tag{12.14}$$

where $k_1, k_2 \in \mathbb{R}$ are strictly positive estimator gains. The estimate $\hat{y}_3(t)$ is generated based on the locally Lipschitz projection defined as [25]

$$\dot{\hat{y}}_3(t) = proj(\hat{y}_3, \phi) = \begin{cases} & \text{if } \underline{y_3} \le \hat{y}_3(t) \le \bar{y}_3 \text{ or} \\ \phi & \text{if } \hat{y}_3(t) > \bar{y}_3 \text{ and } \phi(t) \le 0 \text{ or} \\ & \text{if } \hat{y}_3(t) < \underline{y_3} \text{ and } \phi(t) \ge 0 \\ \bar{\phi} & \text{if } \hat{y}_3(t) > \bar{y}_3 \text{ and } \phi(t) > 0 \\ \breve{\phi} & \text{if } \hat{y}_3(t) < \underline{y_3} \text{ and } \phi(t) < 0 \end{cases} \tag{12.15}$$

where $\phi(y_1, y_2, \hat{y}_3, \omega_1, \omega_2, b_3, e_1, e_2, \dot{e}_1, \dot{e}_2) \in \mathbb{R}$ is defined as

$$\begin{aligned} \phi \triangleq\ & \hat{y}_3^2 b_3 + y_2 \hat{y}_3 \omega_1 - y_1 \hat{y}_3 \omega_2 \\ & + h_1 e_1 + h_2 e_2 + k_3 \frac{h_1(k_1 e_1 + \dot{e}_1) + h_2(k_2 e_2 + \dot{e}_2)}{g}, \end{aligned} \tag{12.16}$$

and $\bar{\phi}(t) \in \mathbb{R}$ and $\breve{\phi}(t) \in \mathbb{R}$ are defined as

$$\bar{\phi} \triangleq \left[1 + \frac{\bar{y}_3 - \hat{y}_3}{\delta}\right] \phi, \quad \breve{\phi} \triangleq \left[1 + \frac{\hat{y}_3 - \underline{y_3}}{\delta}\right] \phi. \tag{12.17}$$

The projection in (12.15) ensures that the estimate $\hat{y}_3 \in \Omega_\delta \; \forall t \geq 0$, where

$$\Omega_\delta = \{y_3 \mid \underline{y_3} - \delta \leq y_3 \leq \bar{y}_3 + \delta\}$$

for some known arbitrary constant $\delta > 0$. In (12.16), $g(y_1, y_2, b) \in \mathbb{R}$ is defined as

$$g(y_1, y_2, b) \triangleq (b_1 - y_1 b_3)^2 + (b_2 - y_2 b_3)^2, \tag{12.18}$$

$h_1(y_1, b_1, b_3)$, $h_2(y_2, b_2, b_3) \in \mathbb{R}$ are defined as

$$h_1(y_1, b_1, b_3) \triangleq (b_1 - y_1 b_3) \qquad h_2(y_2, b_2, b_3) \triangleq (b_2 - y_2 b_3), \tag{12.19}$$

and $k_3(t) \in \mathbb{R}$ is a positive estimator gain defined as

$$k_3(t) > |(2\bar{y}_3 + \delta)|b_3| - y_2 \omega_1 + y_1 \omega_2|. \tag{12.20}$$

Assumption 12.4. The function $g(y_1, y_2, b) \neq 0 \; \forall t \geq 0$. This assumption is an observability condition for (12.13)–(12.15). This condition indicates that the motion of the camera should not be parallel to the optical axis of the camera. If the velocities are parallel to the optical axis of the camera then $\begin{pmatrix} y_1(t) & y_2(t) \end{pmatrix}^T = \begin{pmatrix} \frac{b_1(t)}{b_3(t)} & \frac{b_2(t)}{b_3(t)} \end{pmatrix}^T$, $\forall b_3(t) \neq 0$, which is called *focus of expansion* [12, 20, 5]. This condition also says not all linear velocities can be zero at the same instant of time, *i.e.*, $b_1(t)$, $b_2(t)$, $b_3(t) \neq 0$, simultaneously.

12.4.1.2 Error Dynamics

Differentiating (12.12) and using (12.11)–(12.14) yields

$$\dot{e}_1 = h_1 e_3 - k_1 e_1, \qquad \dot{e}_2 = h_2 e_3 - k_2 e_2. \tag{12.21}$$

To facilitate further development, the expressions in (12.18), (12.19) and (12.21) are used to conclude that

$$e_3 = \frac{h_1(\dot{e}_1 + k_1 e_1) + h_2(\dot{e}_2 + k_2 e_2)}{g}. \tag{12.22}$$

Since $\dot{\hat{y}}_3(t)$ is generated from a projection law, three possible cases for $\dot{e}_3(t)$ are considered.

Case 1: $\underline{y_3} \leq \hat{y}_3(t) \leq \bar{y}_3$ or if $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) \leq 0$ or if $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) \geq 0$: After using (12.11), (12.15), (12.16) and (12.22), the time derivative of $e_3(t)$ in (12.12), can be determined as

$$\dot{e}_3 = f \tag{12.23}$$

where $f(y, \hat{y}_3, b, \omega_1, \omega_2, \bar{e}) \in \mathbb{R}$ is defined as

$$f = (y_3 + \hat{y}_3)b_3 e_3 + (y_2\omega_1 - y_1\omega_2)e_3 - h_1 e_1 - h_2 e_2 - k_3 e_3.$$

Case 2: $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) > 0$: After using (12.11), (12.15), and the definition of $\bar{\phi}(t)$ given by (12.17), the time derivative of $e_3(t)$ in (12.12), can be determined as

$$\dot{e}_3 = \dot{y}_3 - \phi - \frac{\bar{y}_3 - \hat{y}_3}{\delta}\phi = f - \frac{\bar{y}_3 - \hat{y}_3}{\delta}\phi. \tag{12.24}$$

Case 3: $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) < 0$: After using (12.11), (12.15), and the definition of $\breve{\phi}(t)$ given by (12.17), the time derivative of $e_3(t)$ in (12.12), can be determined as

$$\dot{e}_3 = \dot{y}_3 - \phi - \frac{\hat{y}_3 - \underline{y_3}}{\delta}\phi = f - \frac{\hat{y}_3 - \underline{y_3}}{\delta}\phi. \tag{12.25}$$

12.4.1.3 Stability Analysis

The stability of the proposed observer is analyzed in this section using a Lyapunov-based approach. Since the proposed observer uses a projection law, the Lyapunov analysis is examined for three possible cases of projection.

Theorem 12.1. *The observer in (12.13)–(12.15) is exponentially stable provided Assumptions 12.1–12.4 are satisfied, and $k_3(t)$ is selected according to (12.20).*

Proof. Consider a domain $\bar{\mathscr{D}} \subset \mathbb{R}^3$ containing $\bar{e}(0)$ and a continuously differentiable Lyapunov function, $\bar{V}(\bar{e},t) : \bar{\mathscr{D}} \times [0 \times \infty) \to \mathbb{R}^+$, defined as

$$\bar{V}(\bar{e}) \triangleq \frac{1}{2}\bar{e}^T\bar{e}. \tag{12.26}$$

Case 1: $\underline{y_3} \le \hat{y}_3(t) \le \bar{y}_3$ or if $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) \le 0$ or if $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) \ge 0$: Taking the derivative of $\bar{V}(\bar{e})$ and utilizing (12.21) and (12.23) yields

$$\dot{\bar{V}} = -k_1 e_1^2 - k_2 e_2^2 - (-(y_3 + \hat{y}_3)b_3 - y_2\omega_1 + y_1\omega_2 + k_3)e_3^2. \tag{12.27}$$

By choosing the gain $k_3(t)$ according to (12.20), the expression for $\dot{\bar{V}}(t)$ can be upper bounded as

$$\dot{\bar{V}} \le -k_1 e_1^2 - k_2 e_2^2 - k_4 e_3^2 \le -2\min(k_1, k_2, k_4)\bar{V} \tag{12.28}$$

where $k_4 \in \mathbb{R}$ is a strictly positive number. Using the Gronwall–Bellman lemma [7] and (12.26) yields

$$\|\bar{e}(t)\| \le \|\bar{e}(0)\| \exp(-\min(k_1, k_2, k_4)t). \tag{12.29}$$

Case 2: $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) > 0$: Taking the time derivative of $V(\bar{e})$ and utilizing (12.21) and (12.24) yields

$$\dot{\tilde{V}} = -k_1 e_1^2 - k_2 e_2^2 + h_1 e_1 e_3 + h_2 e_2 e_3 + e_3 f - \left[e_3 \frac{\bar{y}_3 - \hat{y}_3}{\delta} \phi \right]. \tag{12.30}$$

Since $\phi(t) > 0$, $e_3(t) < 0$ and $\frac{\bar{y}_3 - \hat{y}_3}{\delta} < 0$, the bracketed term in (12.30) is positive, and (12.30) can be upper bounded as

$$\dot{\tilde{V}} \leq -k_1 e_1^2 - k_2 e_2^2 + h_1 e_1 e_3 + h_2 e_2 e_3 + e_3 f$$
$$\dot{\tilde{V}} \leq -k_1 e_1^2 - k_2 e_2^2 - (-(y_3 + \hat{y}_3) b_3 - y_2 \omega_1 + y_1 \omega_2 + k_3) e_3^2.$$

By choosing the gain $k_3(t)$ according to (12.20), the expression for $\dot{\tilde{V}}(t)$ can be upper bounded as in (12.28), which can be used to obtain (12.29).

Case 3: $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) < 0$: Taking the time derivative of $V(\bar{e})$ and after utilizing (12.21) and (12.25) yields

$$\dot{\tilde{V}} = -k_1 e_1^2 - k_2 e_2^2 + h_1 e_1 e_3 + h_2 e_2 e_3 + e_3 f - \left[e_3 \frac{\hat{y}_3 - \underline{y_3}}{\delta} \phi \right]. \tag{12.31}$$

Since $\phi(t) < 0$, $e_3(t) > 0$ and $\frac{\hat{y}_3 - \underline{y_3}}{\delta} < 0$, the bracketed term in (12.31) is positive, and (12.31) can be upper bounded as

$$\dot{\tilde{V}} \leq -k_1 e_1^2 - k_2 e_2^2 - (-(y_3 + \hat{y}_3) b_3 - y_2 \omega_1 + y_1 \omega_2 + k_3) e_3^2.$$

By choosing the gain $k_3(t)$ according to (12.20), the expression for $\dot{\tilde{V}}(t)$ can be upper bounded as in (12.28), which can be used to obtain (12.29).

The expression in (12.29) indicates that $\bar{e}(t)$ is exponentially stable, and the closed-loop error dynamics can be used to show that all signals remain bounded. Specifically, since $\bar{e}(t) \in \mathcal{L}_\infty$, and $y(t) \in \mathcal{L}_\infty$ from Assumption 12.1, then $\hat{y}(t) \in \mathcal{L}_\infty$. Assumption 12.1–12.2 indicate that $y(t), \omega(t) \in \mathcal{L}_\infty$, so (12.20) can be used to prove that the gain $k_3(t) \in \mathcal{L}_\infty$. Based on the fact that $\bar{e}(t)$, $y(t)$, $\omega(t)$, $b(t)$, $k_3(t) \in \mathcal{L}_\infty$, standard linear analysis methods can be used to prove that $\dot{\bar{e}}(t) \in \mathcal{L}_\infty$. Since $y_3(t)$ is exponentially estimated, (12.3), (12.4), and (12.6) can be used to recover the structure $\bar{m}(t)$ of the feature points. □

12.4.2 Estimation with a Known Linear Velocity

In some scenarios, the linear and angular velocities of the camera may not be completely known (*e.g.*, the camera is attached to a vehicle that does not contain velocity sensors, or the sensor feedback becomes temporarily/permanently lost). In this section, an estimator is designed for the same perspective dynamic system in (12.11), yet the angular velocity is considered unknown and only one of the linear velocities (*i.e.*, b_3) is available. Solving the SaM estimation problem is problematic; hence, some information about the motion of the camera is typically required. To this end,

results such as [6] and [10] and the estimator in this section replace known linear velocity measurements with an uncertain dynamic model of the linear velocity $b(t)$ given by

$$\dot{b_i}(t) = q(y_3 b_i, t) b_i, \forall i = \{1,2,3\}, \tag{12.32}$$

where $q(y_3 b_i, t) \in \mathbb{R}$ is a known function of unknown states.

To facilitate the design and analysis of the subsequent observer, a new state $u(t) \in \mathscr{U} \subset \mathbb{R}^3 \triangleq \left[u_1(t)\ u_2(t)\ u_3(t) \right]^T$ is defined as

$$u_1 = y_3 b_1, \quad u_2 = y_3 b_2, \quad u_3 = b_3, \tag{12.33}$$

where $u_3(t)$ is a measurable linear velocity. The set $\mathscr{U}$ is a closed and bounded set. After utilizing (12.11) and (12.32), the dynamics for $u_1(t)$, $u_2(t)$, $u_3(t)$ can be expressed as

$$\dot{u_1} = y_3 b_3 u_1 + (y_2 \omega_1 - y_1 \omega_2) u_1 + q(u_1) u_1 \tag{12.34}$$

$$\dot{u_2} = y_3 b_3 u_2 + (y_2 \omega_1 - y_1 \omega_2) u_2 + q(u_2) u_2 \tag{12.35}$$

$$\dot{u_3} = q(y_3 b_3, t) b_3. \tag{12.36}$$

The observer in this section is developed to estimate the structure and the partial motion *i.e.*, velocities $b_1(t)$ and $b_2(t)$. Using the same strategy developed for the structure estimator, new estimates $\hat{u}_i(t)\ \forall i = \{1,2,3\} \in \mathbb{R}$ are developed for $u_i(t)$. To quantify this objective, estimate errors $e_4(t)$, $e_5(t)$, $e_6(t) \in \mathbb{R}$ are defined as

$$e_4 = u_1 - \hat{u}_1, \quad e_5 = u_2 - \hat{u}_2, \quad e_6 = u_3 - \hat{u}_3 \tag{12.37}$$

and an augmented error state is defined as $e(t) = \left[\bar{e}^T(t)\ e_4(t)\ e_5(t)\ e_6(t) \right]^T$, where $e_1(t)$, $e_2(t)$, and $e_6(t)$ are measurable.

Assumption 12.5. The functions $q(u_i, t)$ are piecewise differentiable with respect to $u_i(t)$, and the partial derivatives of $q(u_i, t)$ with respect to $u_i(t)$ are bounded, *i.e.* $q(u_i, t)$ is Lipschitz continuous [6, 10].

Assumption 12.6. The function $q(y_3 b_3, t)$ is linear in y_3; hence,

$$q(y_3 b_3, t) - q(\hat{y}_3 b_3, t) = c_1 (y_3 - \hat{y}_3), \tag{12.38}$$

where c_1 is a known scalar where $c_1 \neq 0$, which suggests that the known linear velocity $b_3(t) \neq 0$.

Assumption 12.7. The linear velocities $b_1(t) \neq 0$ and $b_2(t) \neq 0$.

Assumption 12.8. Since the linear camera velocities $b(t)$ are upper and lower bounded by constants, the following inequalities can be determined using Assumption 12.1 and the definition of $u_1(t)$ and $u_2(t)$

$$\underline{u}_1 \leq u_1 \leq \bar{u}_1, \ \ \underline{u}_2 \leq u_2 \leq \bar{u}_2, \ \ \underline{b}_3 \leq b_3 = u_3 \leq \bar{b}_3. \tag{12.39}$$

Remark 12.1. Assumptions 12.6 and 12.7 form observability conditions for the subsequent estimator, *i.e.* camera must have a velocity in all three directions. This observability condition is more stringent than for the estimator designed in Section 12.4.1. It is intuitive that if more parameters have to be estimated, more information is required. The information about $e_3(t)$ is obtained using known $b_3(t)$.

Based on Assumption 12.5, the mean-value theorem can be used to conclude that

$$q(u_1) - q(\hat{u}_1) = J_1 e_4, \qquad q(u_2) - q(\hat{u}_2) = J_2 e_5, \tag{12.40}$$

where $J_1 = \frac{\partial q(u_1)}{\partial u_1} \in \mathbb{R}$, $J_2 = \frac{\partial q(u_2)}{\partial u_2} \in \mathbb{R}$ and are upper bounded as $|J_1| \leq \bar{J}_1$, $|J_2| \leq \bar{J}_2$.

12.4.2.1 Step 1: Angular Velocity Estimation

Solutions are available in literature that can be used to determine the relative angular velocity between the camera and a target. A brief description of the angular velocity estimator presented in [8] is provided as a means to facilitate the subsequent development. The rotation matrices defined in Section 12.2 are related as

$$\bar{R} = R(R^*)^T. \tag{12.41}$$

The relationship between angular velocity of the camera $\omega(t)$ and the rotation matrix $\bar{R}(t)$ is given by [41]

$$[\omega]_\times = \dot{\bar{R}}\bar{R}^T. \tag{12.42}$$

To quantify the rotation mismatch between $\mathscr{F}_c^*$ and $\mathscr{F}_c$, a rotation error vector $e_\omega(t) \in \mathbb{R}^3$ is defined by the angle-axis representation of $\bar{R}(t)$ as

$$e_\omega \triangleq u_\omega(t)\theta_\omega(t), \tag{12.43}$$

where $u_\omega(t) \in \mathbb{R}^3$ represents a unit rotation axis, and $\theta_\omega(t) \in \mathbb{R}$ denotes the rotation angle about $u_\omega(t)$ that is assumed to be confined to region $-\pi < \theta_\omega(t) < \pi$. Taking time derivative of (12.43), the following expression can be obtained

$$\dot{e}_\omega = L_\omega \omega \tag{12.44}$$

where the invertible Jacobian matrix $L_\omega(t) \in \mathbb{R}^{3\times 3}$ is defined as [8]

$$L_\omega \triangleq I_3 - \frac{\theta_\omega}{2}[u_\omega]_\times + \left(1 - \frac{\sin\theta_\omega(t)}{2\sin\left(\frac{\theta_\omega(t)}{2}\right)}\right)[u_\omega]_\times^2. \tag{12.45}$$

A robust integral of the sign of the error (RISE)-based observer $\hat{e}_\omega(t) \in \mathbb{R}^3$ is generated in [8] from the following differential equation

$$\dot{\hat{e}}_\omega = (K_\omega + I_{3\times 3})\tilde{e}_\omega(t) + \int_{t_0}^{t} (K_\omega + I_{3\times 3})\tilde{e}_\omega d\tau + \int_{t_0}^{t} \rho_\omega sgn(\tilde{e}_\omega) d\tau \tag{12.46}$$

where K_ω, $\rho_\omega \in \mathbb{R}^{3\times 3}$ are positive constant diagonal gain matrices, and $\tilde{e}_\omega(t) \in \mathbb{R}^3$ quantifies the observer error as

$$\tilde{e}_\omega(t) \triangleq e_\omega - \hat{e}_\omega.$$

A Lyapunov-based stability analysis is provided in [8] that proves

$$\dot{\hat{e}}_\omega(t) - \dot{e}_\omega(t) \to 0 \qquad \text{as} \qquad t \to \infty, \tag{12.47}$$

and that all closed-loop signals are bounded. Based on (12.44) and (12.47), the angular velocity can be determined as

$$\hat{\omega}(t) = L_\omega^{-1} \dot{\hat{e}}_\omega(t) \qquad \text{as} \qquad t \to \infty. \tag{12.48}$$

An angular velocity estimation error $\tilde{\omega}(t) \in \mathbb{R}^3 \triangleq \left[\, \tilde{\omega}_1(t) \;\; \tilde{\omega}_2(t) \;\; \tilde{\omega}_3(t) \,\right]^T$ is defined as

$$\tilde{\omega}_i(t) = \omega_i(t) - \hat{\omega}_i(t) \;, \quad \forall i = \{1,2,3\}. \tag{12.49}$$

As shown in [8], the angular velocity estimator, given by (12.46), is asymptotically stable, thus the angular velocity estimation error $\|\tilde{\omega}(t)\| \to 0$ as $t \to \infty$.

12.4.2.2 Step 2: Structure Estimation

This section presents an estimator to estimate the structure and motion. One of the linear velocities is assumed to be known and provides the scene scale information to the structure and motion estimator.

The observed states $\hat{y}_1(t)$, $\hat{y}_2(t)$, $\hat{u}_1(t)$, $\hat{u}_2(t)$ and $\hat{u}_3(t)$ are generated according to the update laws

$$\dot{\hat{y}}_1 = \hat{u}_1 - y_1\hat{y}_3 b_3 - y_1 y_2 \hat{\omega}_1 + (1+y_1^2)\hat{\omega}_2 - y_2\hat{\omega}_3 + \rho_1 e_1 \tag{12.50}$$

$$\dot{\hat{y}}_2 = \hat{u}_2 - y_2\hat{y}_3 b_3 - (1+y_2^2)\hat{\omega}_1 + y_1 y_2 \hat{\omega}_2 + y_1\hat{\omega}_3 + \rho_2 e_2 \tag{12.51}$$

$$\begin{aligned} \dot{\hat{u}}_1 \;=\; & \hat{y}_3 b_3 \hat{u}_1 + (y_2\hat{\omega}_1 - y_1\hat{\omega}_2)\hat{u}_1 + q(\hat{u}_1)\hat{u}_1 \\ & + \rho_4(\dot{e}_1 + \rho_1 e_1) + y_1\rho_4 \frac{(\dot{e}_6 + \rho_6 e_6)}{c_1} + e_1 \end{aligned} \tag{12.52}$$

$$\begin{aligned} \dot{\hat{u}}_2 \;=\; & \hat{y}_3 b_3 \hat{u}_2 + (y_2\hat{\omega}_1 - y_1\hat{\omega}_2)\hat{u}_2 + q(\hat{u}_2)\hat{u}_2 \\ & + \rho_5(\dot{e}_2 + \rho_2 e_2) + y_2\rho_5 \frac{(\dot{e}_6 + \rho_6 e_6)}{c_1} + e_2 \end{aligned} \tag{12.53}$$

$$\dot{\hat{u}}_3 = q(\hat{y}_3 b_3, t) b_3 + \rho_6 e_6 \tag{12.54}$$

where $\rho_i \in \mathbb{R}$,$\forall i = \{1,...,6\}$ are observer gains, $c_1 \in \mathbb{R}$ is the known, nonzero constant introduced in (12.38) and $\hat{\omega}_i(t)$ are given by (12.48). The estimate $\hat{y}_3(t)$ is generated based on the locally Lipschitz projection defined in (12.15) where $\phi(\cdot)$ is now defined as

$$\phi \triangleq \hat{y}_3^2 b_3 + y_2 \hat{y}_3 \hat{\omega}_1 - y_1 \hat{y}_3 \hat{\omega}_2 - y_1 b_3 e_1 - y_2 b_3 e_2 + c_1 b_3 e_6 + \rho_3 \frac{(\dot{e}_6 + \rho_6 e_6)}{c_1 b_3}. \quad (12.55)$$

In (12.50)–(12.55) the observer gains ρ_1, ρ_2 and $\rho_6 \in \mathbb{R}$ are strictly positive constants, and the gains $\rho_3(t)$, $\rho_4(t)$, and $\rho_5(t)$ are defined as

$$\rho_3(t) > |(2\bar{y}_3 + \delta)|b_3| + |b_3|(\bar{u}_1 + \bar{u}_2) + y_2\omega_1 - y_1\omega_2|, \quad (12.56)$$

$$\rho_4(t) > |(\bar{y}_3 + \delta)|b_3| + |b_3|\bar{u}_1 + \bar{u}_1\bar{J}_4 + q(\hat{u}_1) + y_2\omega_1 - y_1\omega_2|, \quad (12.57)$$

$$\rho_5(t) > |(\bar{y}_3 + \delta)|b_3| + |b_3|\bar{u}_2 + \bar{u}_2\bar{J}_5 + q(\hat{u}_1) + y_2\omega_1 - y_1\omega_2|. \quad (12.58)$$

12.4.2.3 Error Dynamics

Differentiating (12.12) and (12.37), and using (12.50)–(12.54) yields the following closed-loop error dynamics

$$\dot{e}_1 = e_4 - y_1 b_3 e_3 - \rho_1 e_1 - y_1 y_2 \tilde{\omega}_1 + (1 + y_1^2)\tilde{\omega}_2 - y_2 \tilde{\omega}_3, \quad (12.59)$$

$$\dot{e}_2 = e_5 - y_2 b_3 e_3 - \rho_2 e_2 - (1 + y_2^2)\tilde{\omega}_1 + y_1 y_2 \tilde{\omega}_2 + y_1 \tilde{\omega}_3, \quad (12.60)$$

$$\begin{aligned}\dot{e}_4 = {} & b_3 u_1 e_3 + \hat{y}_3 b_3 e_4 + (y_2\omega_1 - y_1\omega_2)e_4 + y_2\hat{u}_1\tilde{\omega}_1 - y_1\hat{u}_1\tilde{\omega}_2 \\ & + u_1 J_4 e_4 + q(\hat{u}_1)e_4 - \rho_4(\dot{e}_1 + \rho_1 e_1 + y_1 b_3(\frac{\dot{e}_6 + \rho_6 e_6}{c_1 b_3})) - e_1,\end{aligned}$$

$$\begin{aligned}\dot{e}_5 = {} & b_3 u_2 e_3 + \hat{y}_3 b_3 e_5 + (y_2\omega_1 - y_1\omega_2)e_5 + y_2\hat{u}_2\tilde{\omega}_1 - y_1\hat{u}_2\tilde{\omega}_2 \\ & + u_2 J_5 e_5 + q(\hat{u}_2)e_5 - \rho_5(\dot{e}_2 + \rho_2 e_2 + y_2 b_3(\frac{\dot{e}_6 + \rho_6 e_6}{c_1 b_3})) - e_2,\end{aligned}$$

$$\dot{e}_6 = c_1 b_3 e_3 - \rho_6 e_6. \quad (12.61)$$

By defining auxiliary signals $\Psi_1(t) \in \mathbb{R}$ and $\Psi_2(t) \in \mathbb{R}$ as

$$\Psi_1 \triangleq -y_1 y_2 \tilde{\omega}_1 + (1 + y_1^2)\tilde{\omega}_2 - y_2 \tilde{\omega}_3, \quad (12.62)$$

$$\Psi_2 \triangleq -(1 + y_2^2)\tilde{\omega}_1 + y_1 y_2 \tilde{\omega}_2 + y_1 \tilde{\omega}_3, \quad (12.63)$$

(12.59) and (12.60) can be re-expressed as

$$\dot{e}_1 = e_4 - y_1 b_3 e_3 - \rho_1 e_1 + \Psi_1, \quad (12.64)$$

$$\dot{e}_2 = e_5 - y_2 b_3 e_3 - \rho_2 e_2 + \Psi_2, \quad (12.65)$$

and, after using (12.61), $e_3(t)$ can be expressed as

$$e_3 = \frac{\dot{e}_6 - \rho_6 e_6}{c_1 b_3}. \tag{12.66}$$

The results from the angular velocity estimator prove that $\tilde{\omega}_1, \tilde{\omega}_2, \tilde{\omega}_3 \to 0$, therefore $\Psi_1(t), \Psi_2(t) \to 0$ as $t \to \infty$. By using the expressions (12.64)–(12.66) and the definitions of $\Psi_1(t)$ and $\Psi_2(t)$, the expressions for $e_4(t)$ and $e_5(t)$ can be obtained as

$$e_4 = \dot{e}_1 + \rho_1 e_1 + y_1 \frac{\dot{e}_6 + \rho_6 e_6}{c_1} - \Psi_1, \tag{12.67}$$

$$e_5 = \dot{e}_2 + \rho_2 e_2 + y_2 \frac{\dot{e}_6 + \rho_6 e_6}{c_1} - \Psi_2. \tag{12.68}$$

Utilizing (12.62), (12.63), (12.67) and (12.68), the error dynamics for $e_4(t)$, $e_5(t)$, and $e_6(t)$ can be written as

$$\begin{aligned} \dot{e}_4 &= b_3 u_1 e_3 + \hat{y}_3 b_3 e_4 + (y_2\omega_1 - y_1\omega_2)e_4 + y_2\hat{u}_1\tilde{\omega}_1 - y_1\hat{u}_1\tilde{\omega}_2 \\ &\quad + u_1 J_4 e_4 + q(\hat{u}_1)e_4 - \rho_4(e_4 + \Psi_1) - e_1, \end{aligned} \tag{12.69}$$

$$\begin{aligned} \dot{e}_5 &= b_3 u_2 e_3 + \hat{y}_3 b_3 e_5 + (y_2\omega_1 - y_1\omega_2)e_5 + y_2\hat{u}_2\tilde{\omega}_1 - y_1\hat{u}_2\tilde{\omega}_2 \\ &\quad + u_2 J_5 e_5 + q(\hat{u}_2)e_5 - \rho_5(e_5 + \Psi_2) - e_2, \end{aligned} \tag{12.70}$$

$$\dot{e}_6 = c_1 b_3 e_3 - \rho_6 e_6. \tag{12.71}$$

Since $\dot{\hat{y}}_3(t)$ is generated from a projection law, three possible cases for $\dot{e}_3(t)$ are considered.

Case 1: $\underline{y_3} \le \hat{y}_3(t) \le \bar{y}_3$ or if $\hat{y}_3(t) > \bar{y}_3$ and $\beta(t) \le 0$ or if $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) \ge 0$: After using (12.15), (12.55) and (12.66), the time derivative of $e_3(t)$ in (12.12), can be determined as

$$\dot{e}_3 = \alpha \tag{12.72}$$

where $\alpha(y, b_3, \omega_1, \omega_2, e_3, e_6, \dot{e}_6) \in \mathbb{R}$ is defined as

$$\begin{aligned} \alpha &= (y_3 + \hat{y}_3)b_3 e_3 + (y_2\omega_1 - y_1\omega_2)e_3 + y_2\hat{y}_3\tilde{\omega}_1 - y_1\hat{y}_3\tilde{\omega}_2 \\ &\quad - \rho_3 e_3 + y_1 b_3 e_1 + y_2 b_3 e_2 - c_1 b_3 e_6. \end{aligned} \tag{12.73}$$

Case 2: $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) > 0$: After using (12.55), and the definition of $\bar{\phi}(t)$ given by (12.17), the time derivative of $e_3(t)$ in (12.12), can be determined as

$$\dot{e}_3 = \dot{y}_3 - \phi - \frac{\bar{y}_3 - \hat{y}_3}{\delta}\phi = \alpha - \frac{\bar{y}_3 - \hat{y}_3}{\delta}\phi. \tag{12.74}$$

Case 3: $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) < 0$: After using (12.55), and the definition of $\breve{\phi}(t)$ given by (12.17), the time derivative of $e_3(t)$ in (12.12), can be determined as

$$\dot{e}_3 = \dot{y}_3 - \phi - \frac{\hat{y}_3 - \underline{y_3}}{\delta}\phi = \alpha - \frac{\hat{y}_3 - \underline{y_3}}{\delta}\phi. \tag{12.75}$$

12.4.2.4 Stability Analysis

The stability of the proposed observer is analyzed in this section using a Lyapunov-based approach. Since the proposed observer uses a projection law, the Lyapunov analysis is examined for three possible cases of projection.

Theorem 12.2. *The observer in (12.50)–(12.54) is asymptotically stable provided Assumptions 12.1–12.8 are satisfied, and $\rho_3(t)$, $\rho_4(t)$ and $\rho_5(t)$ are selected according to (12.56)–(12.58).*

Proof. Consider a domain $\mathscr{D} \subset \mathbb{R}^6$ containing $e(0)$ and a continuously differentiable Lyapunov function, $V(e,t) : \mathscr{D} \times [0,\infty) \to \mathbb{R}^+$, defined as

$$V(e) \triangleq \frac{1}{2} e^T e. \tag{12.76}$$

Case 1: $\underline{y_3} \leq \hat{y}_3(t) \leq \bar{y}_3$ or if $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) \leq 0$ or if $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) \geq 0$: After utilizing error dynamics in (12.64), (12.65), (12.69)–(12.72), the time derivative of (12.76) can be expressed as

$$\begin{aligned}
\dot{V} = {} & e_1(e_4 - y_1 b_3 e_3 - \rho_1 e_1 + \Psi_1) + e_2(e_5 - y_2 b_3 e_3 - \rho_2 e_2 + \Psi_2) \\
& + e_3((y_3 + \hat{y}_3) b_3 e_3 + (y_2\omega_1 - y_1\omega_2) e_3 + y_2 \hat{y}_3 \tilde{\omega}_1 - y_1 \hat{y}_3 \tilde{\omega}_2 \\
& - \rho_3 e_3 + y_1 b_3 e_1 + y_2 b_3 e_2 - c_1 b_3 e_6) \\
& + e_4(b_3 u_1 e_3 + \hat{y}_3 b_3 e_4 + (y_2\omega_1 - y_1\omega_2) e_4 + y_2 \hat{u}_1 \tilde{\omega}_1 - y_1 \hat{u}_1 \tilde{\omega}_2 \\
& + u_1 J_4 e_4 + q(\hat{u}_1) e_4 - \rho_4(e_4 + \Psi_1) - e_1) \\
& + e_5(b_3 u_2 e_3 + \hat{y}_3 b_3 e_5 + (y_2\omega_1 - y_1\omega_2) e_5 + y_2 \hat{u}_2 \tilde{\omega}_1 - y_1 \hat{u}_2 \tilde{\omega}_2 \\
& + u_2 J_5 e_5 + q(\hat{u}_2) e_5 - \rho_5(e_5 + \Psi_2) - e_2) \\
& + e_6(c_1 b_3 e_3 - \rho_6 e_6).
\end{aligned}$$

Using the facts that

$$|e_3||e_4| \leq \frac{1}{2} e_3^2 + \frac{1}{2} e_4^2, \qquad |e_3||e_5| \leq \frac{1}{2} e_3^2 + \frac{1}{2} e_5^2$$

along with the bounds on the terms $u_1(t)$, $u_2(t)$, J_4, J_5, and re-arranging the terms, the following inequality for $\dot{V}(t)$ can be developed

$$\begin{aligned}
\dot{V} \leq {} & -\rho_1 e_1^2 - \rho_2 e_2^2 - \rho_6 e_6^2 \\
& - (-(y_3 + \hat{y}_3) b_3 - y_2\omega_1 + y_1\omega_2 - |b_3| \bar{u}_1 - |b_3| \bar{u}_2 + \rho_3) e_3^2 \\
& - (-\hat{y}_3 b_3 - y_2\omega_1 + y_1\omega_2 - \bar{u}_1 \bar{J}_4 - p(\hat{u}_1) - |b_3| \bar{u}_1 + \rho_4) e_4^2 \\
& - (-\hat{y}_3 b_3 - y_2\omega_1 + y_1\omega_2 - \bar{u}_2 \bar{J}_5 - p(\hat{u}_2) - |b_3| \bar{u}_2 + \rho_5) e_5^2 \\
& + (e_1 - \rho_4 e_4) \Psi_1 + (e_2 - \rho_5 e_5) \Psi_2 \\
& + (y_2 \hat{y}_3 e_3 + y_2 \hat{u}_1 e_4 + y_2 \hat{u}_2 e_5) \tilde{\omega}_1 - (y_1 \hat{y}_3 e_3 + y_1 \hat{u}_1 e_4 + y_1 \hat{u}_2 e_5) \tilde{\omega}_2.
\end{aligned}$$

Choosing gains $\rho_3(t)$, $\rho_4(t)$, $\rho_5(t)$ as given by (12.56), (12.57) and (12.58) yields

$$\dot{V} \leq -\rho_1 e_1^2 - \rho_2 e_2^2 - \rho_7 e_3^2 - \rho_8 e_4^2 - \rho_9 e_5^2 - \rho_6 e_6^2 + \Pi \tag{12.77}$$

where $\rho_i \in \mathbb{R}, \forall i = \{7,8,9\}$ are strictly positive constants, and $\Pi(t) \in \mathbb{R}$ is

$$\begin{aligned}\Pi(t) \triangleq\ & (e_1 - \rho_4 e_4)\Psi_1 + (e_2 - \rho_5 e_5)\Psi_2 + (y_2 \hat{y}_3 e_3 + y_2 \hat{u}_1 e_4 + y_2 \hat{u}_2 e_5)\tilde{\omega}_1 \\ & -(y_1 \hat{y}_3 e_3 + y_1 \hat{u}_1 e_4 + y_1 \hat{u}_2 e_5)\tilde{\omega}_2.\end{aligned} \tag{12.78}$$

Thus, (12.77) can be upper bounded as

$$\dot{V} \leq -\lambda V + \Pi. \tag{12.79}$$

Case 2: $\hat{y}_3(t) > \bar{y}_3$ and $\phi(t) > 0$. Taking the time derivative of $V(e)$ and utilizing (12.64), (12.65), (12.69)–(12.71), (12.74) yields

$$\begin{aligned}\dot{V} =\ & e_1(e_4 - y_1 b_3 e_3 - \rho_1 e_1 + \Psi_1) + e_2(e_5 - y_2 b_3 e_3 - \rho_2 e_2 + \Psi_2) + \alpha e_3 \\ & - \left[e_3 \frac{\bar{y}_3 - \hat{y}_3}{\delta} \phi \right] + e_4 [b_3 u_1 e_3 + \hat{y}_3 b_3 e_4 + (y_2 \omega_1 - y_1 \omega_2) e_4 + y_2 \hat{u}_1 \tilde{\omega}_1 \\ & - y_1 \hat{u}_1 \tilde{\omega}_2 + u_1 J_4 e_4 + q(\hat{u}_1) e_4 - \rho_4(e_4 + \Psi_1) - e_1] \\ & + e_5 [b_3 u_2 e_3 + \hat{y}_3 b_3 e_5 + (y_2 \omega_1 - y_1 \omega_2) e_5 + y_2 \hat{u}_2 \tilde{\omega}_1 - y_1 \hat{u}_2 \tilde{\omega}_2 \\ & + u_2 J_5 e_5 + q(\hat{u}_2) e_5 - \rho_5(e_5 + \Psi_2) - e_2] \\ & + e_6(c_1 b_3 e_3 - \rho_6 e_6).\end{aligned} \tag{12.80}$$

Since $\phi(t) > 0$, $e_3(t) < 0$ and $\frac{\bar{y}_3 - \hat{y}_3}{\delta} < 0$, the bracketed term in (12.80) is strictly positive, and the expression for $\dot{V}(t)$ can be upper bounded as in (12.77).

Case 3: $\hat{y}_3(t) < \underline{y_3}$ and $\phi(t) < 0$. Taking the time derivative of $V(e)$ and utilizing (12.64), (12.65), (12.69)–(12.71), (12.75) yields

$$\begin{aligned}\dot{V} =\ & e_1(e_4 - y_1 b_3 e_3 - \rho_1 e_1 + \Psi_1) + e_2(e_5 - y_2 b_3 e_3 - \rho_2 e_2 + \Psi_2) + \alpha e_3 \\ & - \left[e_3 \frac{\bar{y}_3 - \hat{y}_3}{\delta} \phi \right] + e_4 [b_3 u_1 e_3 + \hat{y}_3 b_3 e_4 + (y_2 \omega_1 - y_1 \omega_2) e_4 + y_2 \hat{u}_1 \tilde{\omega}_1 \\ & - y_1 \hat{u}_1 \tilde{\omega}_2 + u_1 J_4 e_4 + q(\hat{u}_1) e_4 - \rho_4(e_4 + \Psi_1) - e_1] \\ & + e_5 [b_3 u_2 e_3 + \hat{y}_3 b_3 e_5 + (y_2 \omega_1 - y_1 \omega_2) e_5 + y_2 \hat{u}_2 \tilde{\omega}_1 - y_1 \hat{u}_2 \tilde{\omega}_2 \\ & + u_2 J_5 e_5 + q(\hat{u}_2) e_5 - \rho_5(e_5 + \Psi_2) - e_2] \\ & + e_6(c_1 b_3 e_3 - \rho_6 e_6).\end{aligned}$$

Since $\phi(t) < 0$, $e_3(t) > 0$ and $\frac{\bar{y}_3 - \hat{y}_3}{\delta} < 0$, the bracketed term in (12.80) is strictly positive, and the expression for $\dot{V}(t)$ can be upper bounded as in (12.77).

The results in Section 12.4.2.1 indicate that $\|\tilde{\omega}(t)\| \to 0$ as $t \to \infty$, and hence, $\Psi_1(t), \Psi_2(t)$, $\Pi(t) \to 0$ as $t \to \infty$. Thus, the expression (12.79) is a linear system

with vanishing perturbations. The nominal system $\dot{V} \leq -\lambda V$ is exponentially stable. Thus, it can be proved that the system is asymptotically stable [26] *i.e.* $V(e) \to 0$ as $t \to \infty$. Thus, the definition of $V(e)$ can be used to conclude that $\|e(t)\| \to 0$ as $t \to \infty$. Since the system is asymptotically stable, $V(e) \in L_\infty$; hence $e(t) \in L_\infty$. Since $e(t) \in L_\infty$, and using Assumptions 12.1 and 12.7, $y(t), u(t) \in L_\infty$, thus $\hat{y}(t), \hat{u}(t) \in L_\infty$. Since $e(t)$, $y(t)$, $u(t)$, $\omega(t)$, $b(t) \in L_\infty$, a linear analysis proves that $\dot{e}(t) \in L_\infty$. Also, the gains $\rho_3(t)$, $\rho_4(t)$, $\rho_5(t) \in L_\infty$. Hence, all the signal are bounded and the proposed estimator identifies/estimates the states asymptotically. As $y_3(t)$, $u_1(t)$, and $u_2(t)$ can be estimated, it is possible to partially recover the motion parameters, *i.e.* $b_1(t)$ and $b_2(t)$. □

References

[1] Azarbayejani, A., Pentland, A.P.: Recursive estimation of motion, structure, and focal length. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(6), 562–575 (1995)

[2] Bartoli, A., Sturm, P.: Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. International Journal of Computer Vision 52(1), 45–64 (2003)

[3] Bartoli, A., Sturm, P.: Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. Computer Vision and Image Understanding 100(3), 416–441 (2005)

[4] Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)

[5] Chen, X., Kano, H.: A new state observer for perspective systems. IEEE Trans. Automat. Contr. 47(4), 658–663 (2002)

[6] Chen, X., Kano, H.: State observer for a class of nonlinear systems and its application to machine vision. IEEE Trans. Automat. Contr. 49(11), 2085–2091 (2004)

[7] Chicone, C.: Ordinary Differential Equations with Applications, 2nd edn. Springer, Heidelberg (2006)

[8] Chitrakaran, V., Dawson, D.M., Dixon, W.E., Chen, J.: Identification a moving object´s velocity with a fixed camera. Automatica 41(3), 553–562 (2005)

[9] Chiuso, A., Favaro, P., Jin, H., Soatto, S.: Structure from motion causally integrated over time. IEEE Trans. Pattern Anal. Machine Intell. 24(4), 523–535 (2002)

[10] Dahl, O., Nyberg, F., Heyden, A.: Nonlinear and adaptive observers for perspective dynamic systems. In: American Controls Conference, New York City, USA, pp. 966–971 (2007)

[11] De Luca, A., Oriolo, G., Giordano, P.: On-line estimation of feature depth for image-based visual servoing schemes. In: Proc. IEEE Int. Conf. Robotics and Automation, pp. 2823–2828 (2007)

[12] Dixon, W.E., Fang, Y., Dawson, D.M., Flynn, T.J.: Range identification for perspective vision systems. IEEE Trans. Automat. Contr. 48(12), 2232–2238 (2003)

[13] Faugeras, O.D., Lustman, F.: Motion and structure from motion in a piecewise planar environment. Int. J. Pattern Recog. and Artificial Intell. 2(3), 485–508 (1988)

[14] Ghosh, B., Loucks, E.: A perspective theory for motion and shape estimation in machine vision. SIAM J. on Control and Optimization 33(5), 1530–1559 (1995)

[15] Ghosh, B.K., Inaba, H., Takahashi, S.: Identification of riccati dynamics under perspective and orthographic observations. IEEE Transactions on Automatic Control 45(7), 1267–1278 (2000)
[16] Hartley, R., Zisserman, A.: Multiple View Geometry. Cambridge University Press, Cambridge (2003)
[17] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
[18] Hu, G., Aiken, D., Gupta, S., Dixon, W.: Lyapunov-Based Range Identification For Paracatadioptric Systems. IEEE Transactions on Automatic Control 53(7), 1775–1781 (2008)
[19] Hutchinson, S., Hager, G., Corke, P.: A tutorial on visual servo control. IEEE Trans. Robot. Automat. 12(5), 651–670 (1996)
[20] Jankovic, M., Ghosh, B.: Visually guided ranging from observations points, lines and curves via an identifier based nonlinear observer. Systems and Control Letters 25(1), 63–73 (1995)
[21] Kahl, F.: Multiple view geometry and the l_∞-norm. In: International Conference on Computer Vision, pp. 1002–1009 (2005)
[22] Kahl, F., Hartley, R.: Multiple-view geometry under the l_∞-norm. IEEE Transactions on Patterm Analysis and Machine Intelligence 30(9), 1603–1617 (2008)
[23] Kano, H., Ghosh, B.K., Kanai, H.: Single camera based motion and shape estimation using extended kalman filtering. Mathematical and Computer Modelling 34, 511–525 (2001)
[24] Karagiannis, D., Astolfi, A.: A new solution to the problem range identification in perspective vision systems. IEEE Trans. Automat. Contr. 50, 2074–2077 (2005)
[25] Khalil, H.: Adaptive Output Feedback Control of Nonlinear Systems Represented by Input-Output Models. IEEE Transactions on Automatic Control 41(2), 177 (1996)
[26] Khalil, H.K.: Nonlinear Systems, 3 edn. Prentice Hall, New Jersey (2002)
[27] Lowe, D.G.: Distinctive image feature from scale-invariant keypoints. Int. J. Computer Vision 60, 91–110 (2004)
[28] Ma, L., Cao, C., Hovakimayan, N., Dixon, W.E., Woolsey, C.: Range identification in the presence of unknown motion parameters for perspective vision systems. In: American Controls Conference, New York City, USA, pp. 972–977 (2007)
[29] Ma, Y., Soatto, S., Kosecká, J., Sastry, S.: An Invitation to 3-D Vision. Springer, Heidelberg (2004)
[30] Matthies, L., Kanade, T., Szeliski, R.: Kalman filter-based algorithms for estimating depth from image sequences. Int. J. Computer Vision 3, 209–236 (1989)
[31] Oliensis, J.: A critique of structure-from-motion algorithms. Computer Vision and Image Understanding 80, 172–214 (2000)
[32] Oliensis, J.: Exact two-image structure from motion. IEEE Transactions on Patterm Analysis and Machine Intelligence 24(12), 1618–(2002)
[33] Oliensis, J., Hartley, R.: Iterative extensions of the strum/triggs algorithm: convergence and nonconvergence. IEEE Transactions on Patterm Analysis and Machine Intelligence 29(12), 2217–2233 (2007)
[34] Quian, G., Chellappa, R.: Structure from motion using sequential monte carlo methods. International Journal of Computer Vision 59, 5–31 (2004)
[35] Ramalingam, S., Lodha, S., Sturm, P.: A generic structure-from-motion framework. Computer Vision and Image Understanding 103(3), 218–228 (2006)

[36] Ribnick, E., Atev, S., Papanikolopoulos, N.: Estimating 3D Positions and Velocities of Projectiles from Monocular Views. IEEE transactions on pattern analysis and machine intelligence 31(5), 938 (2009)
[37] Sim, K., Hartley, R.: Recovering camera motion using l_∞ minimization. In: Computer Vision and Pattern Recognition, vol. 1, pp. 1230–1237 (2006)
[38] Soatto, S.: 3d structure from visual motion: Modeling, representation and observability. Automatica 33(7), 1287–1312 (1997)
[39] Soatto, S., Frezza, R., Perona, P.: Motion estimation via dynamic vision. IEEE Trans. Automat. Contr. 41(3), 393–413 (1996)
[40] Soatto, S., Perona, P.: Reducing structure from motion: A general framework for dynamic vision, part 1: Modeling. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(9) (1998)
[41] Spong, M.W., Vidyasagar, M.: Robot Dynamics and Control. Wiley, New York (1989)
[42] Sturm, P., Triggs, B.: A factorization based algorithm for multi-image projective structure and motion. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 709–720. Springer, Heidelberg (1996)
[43] Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. rep., Carnegie Mellon University (1991)
[44] Zhang, Z., Hanson, A.: 3D reconstruction based on homography mapping. In: Proc. ARPA Image Understanding Workshop Palm Springs, CA (1996)

Chapter 13
Visual Servoing and Pose Estimation with Cameras Obeying the Unified Model

Omar Tahri, Youcef Mezouar, François Chaumette, and Helder Araujo

Abstract. In this chapter, both visual servoing and pose estimation from a set of points are dealt with. More precisely, a unique scheme based on the projection onto the unit sphere for cameras obeying the unified model is proposed. From the projection onto the surface of the unit sphere, new visual features based on invariants to rotations are proposed. It is shown that satisfactory results can be obtained using these features for visual servoing and pose estimation as well.

13.1 Introduction

Visual servoing aims at controlling robotic systems by the information provided by one or more cameras. According to the space where the visual features are defined, several kinds of visual servoing can be distinguished. In position-based visual servoing (PBVS) [33, 21], the features are defined in the 3D space. An adequate 3D trajectory is usually obtained using PBVS, such as a geodesic for the orientation and a straight line for the translation. However, position-based visual servoing may suffer from potential instabilities due to image noise [5]. Furthermore, the knowledge of an exact object 3D model is required. On the contrary, in image-based visual servo (IBVS) [9], the robot motions are controlled by canceling errors on visual features defined in the image. This kind of visual-servo is more robust to image noise and calibration errors than PBVS, in general. However, as soon as the initial error between the initial and the desired poses is large, the 3D behavior becomes

Omar Tahri and Helder Araujo
Institute for Systems and Robotics, Polo II 3030-290 Coimbra, Portugal
e-mail: {omartahri,helder}@isr.uc.pt

Youcef Mezouar
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France
e-mail: mezouar@univ-bpclermont.fr

François Chaumette
INRIA, Campus de Beaulieu, 35042 Rennes, France
e-mail: francois.chaumette@irisa.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 231–250.
springerlink.com

unpredictable when the visual features are not adequately chosen. Furthermore, other problems may appear such as reaching local minimum or a task singularity [5]. A compromise and an hybrid visual servoing [19] can be obtained by combining features in image and partial 3D data.

In this chapter, we are concerned with IBVS. In fact, the main cause of trouble for IBVS is the strong nonlinearities in the relation from the image space to the workspace that are generally observed in the interaction matrix. In principle, an exponential decoupled decrease will be obtained simultaneously on the visual features and on the camera velocity, which would provide a perfect behavior, if the interaction matrix is constant. Unfortunately, that is not usually the case. To overcome the problem of nonlinearities observed in the interaction matrix, an approach consists in using the measures to build particular visual features that will ensure expected properties of the control scheme. In fact, the way to design adequate visual features is directly linked to the modeling of their interaction with the robot motion, from which all control properties can be analyzed theoretically. If the interaction is too complex (*i.e.* highly nonlinear and coupled), the analysis becomes impossible. Several works have been realized in IBVS following this general objective. In [24], a vanishing point and the horizon line have been selected. This choice ensures a good decoupling between translational and rotational degrees of freedom (DOF). In [15], vanishing points have also been used for a dedicated object (a 3D rectangle), once again for decoupling properties. For the same object, six visual features have been designed in [6] to control the 6 DOF of a robot arm, following a partitioned approach. In [14], the coordinates of points are expressed in a cylindrical coordinate system instead of the classical Cartesian one, so as to improve the robot trajectory. In [13], the three coordinates of the centroid of an object in a virtual image obtained through a spherical projection have been selected to control 3 DOF of an underactuated system. Recently, [11] proposed a decoupled visual servoing from spheres using a spherical projection model. Despite of the large quantity of results obtained in the last few years, the choice of the set of visual features to be used in the control scheme is still an open question in terms of stability analysis and validity for different kinds of sensor and environment.

In this chapter, invariants computed from the projection onto the surface of the unit sphere will be used to improve the IBVS behavior in terms of convergence domain and 3D behavior. In previous works, the invariance property of some combinations of image moments computed from image regions or a set of points have been used to decouple the DOF from each-other. For instance, in [28, 30], moments allow using of intuitive geometrical features, such as the center of gravity or the orientation of an object. However, these works only concerned planar objects and conventional perspective cameras. More recently, a new decoupled IBVS from the projection onto the unit sphere has been proposed in [31]. The proposed method is based on polynomials invariant to rotational motion computed from a set of image points. This current work improves the proposed features given in [31]. More precisely, the new features allow obtaining interaction matrices almost constant with respect to the depth distributions. This decreases the system nonlinearity and improves the convergence speed and rate.

The second part of this chapter deals with the pose estimation problem. There are many applications of pose estimation, where the 6 parameters of the camera pose have to be calculated from known correspondences with known scene structure: robot localization using a vision sensor, or PBVS [33]. The pose estimation is one of most classical problem in vision [7, 16]. This problem is more than 150 years old and there is recent renewed interest because of automated navigation and model-based vision systems. Numerous methods have been proposed in the literature and giving an exhaustive list of them is certainly impossible. The pose estimation methods can be divided into several categories according to the used features, direct methods or iterative methods. The geometric features considered for the estimation of the pose are often points [7], segments [8], contours, conics [25] or image moments [29]. Another important issue is the registration problem. Purely geometric [8], or numerical and iterative [7] approaches may be considered. Linear approaches are suitable for real-time applications and give closed-form solutions free of initialization [10, 1]. Full-scale nonlinear optimization techniques [17] consist of minimizing the error between the observation and the projection of the model. The main advantage of these approaches is their accuracy. The main drawback is that they may be subject to local minima and, worse, divergence.

The method we propose in this chapter is based on virtual visual servoing (VVS) using moment invariants as features. In other words, we consider the problem of the pose computation as similar to the positioning of a virtual camera using features in the image [26, 20]. This method is equivalent to nonlinear methods that consist in minimizing a cost function using iterative algorithms. The main idea behind the method we propose is based on the following fact: the features that can be used for visual servoing to ensure a large convergence domain and adequate 3D behavior can be used to obtain a large convergence domain and high convergence speed for pose estimation using VVS.

As mentioned above, the features we propose are computed from the projection onto the unit sphere. This means that the proposed method can be applied not only to conventional cameras but also to all omnidirectional cameras obeying the unified model [12, 4]. Omnidirectional cameras are usually intended as a vision system providing a 360^o panoramic view of the scene. Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by simply combining mirrors and conventional cameras, or employing purely dioptric fisheye lenses [3]. In practice, it is highly desirable that such imaging systems have a single viewpoint [3, 27]. That is, there exists a single center of projection, so that, every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction. The reason why a single viewpoint is so desirable is that it permits the extension of several results obtained for conventional cameras. The pose estimation method we propose is thus valid for catadioptric, conventional and some fisheye cameras.

This chapter is organized as follows:

- in the next section, the unified camera model is recalled;
- in Section 13.3, the theoretical background of this work is detailed;

- in Section 13.4, the feature choice to control the 6 DOF of the camera or to estimate its pose is explained; and
- in Section 13.5, validations results for visual servoing and pose estimation are presented. In this way, the pose estimation method using VVS is compared to linear pose estimation method [1] and an iterative method [2].

13.2 Camera Model

Central imaging systems can be modeled using two consecutive projections: spherical then perspective. This geometric formulation called the *unified model* was proposed by Geyer and Daniilidis in [12]. Consider a virtual unitary sphere centered on C_m and the perspective camera centered on C_p (refer to Figure 13.1). The frames attached to the sphere and the perspective camera are related by a simple translation of $-\xi$ along the Z-axis. Let $\mathscr{X}$ be a 3D point with coordinates $\mathscr{X} = (X, Y, Z)^\top$ in $\mathscr{F}_m$. The world point $\mathscr{X}$ is projected onto the image plane at a point with homogeneous coordinates $\mathbf{p} = \mathbf{K}\mathbf{m}$, where $\mathbf{K}$ is a 3×3 upper triangular matrix containing the conventional camera intrinsic parameters coupled with mirror intrinsic parameters and

$$\mathbf{m} = \left(x,\, y,\, 1\right)^\top = \left(\frac{X}{Z+\xi\|\mathscr{X}\|},\, \frac{Y}{Z+\xi\|\mathscr{X}\|},\, 1\right)^\top. \tag{13.1}$$

The matrix $\mathbf{K}$ and the parameter ξ can be obtained after calibration using, for example, the methods proposed in [22]. In the sequel, the imaging system is assumed to be calibrated. In this case, the inverse projection onto the unit sphere can be obtained by

$$\mathscr{X}_\mathbf{s} = \lambda \left(x,\, y,\, 1 - \frac{\xi}{\lambda}\right)^\top, \tag{13.2}$$

where $\lambda = \frac{\xi + \sqrt{1+(1-\xi^2)(x^2+y^2)}}{1+x^2+y^2}$.

Note that the conventional perspective camera is nothing but a particular case of this model (when $\xi = 0$). The projection onto the unit sphere from the image plane is possible for all sensors obeying the unified model.

13.3 Mathematical Background

This section first introduces pose estimation via VVS, and then moments from points projected onto the unit sphere.

13.3.1 Visual Servoing and Pose Estimation

In few words, we recall that the time variation $\dot{\mathbf{s}}$ of the visual features $\mathbf{s}$ can be expressed linearly with respect to the relative camera-object kinematics screw $\mathbf{V} = (\mathbf{v}, \boldsymbol{\omega})$:

$$\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\mathbf{V}, \tag{13.3}$$

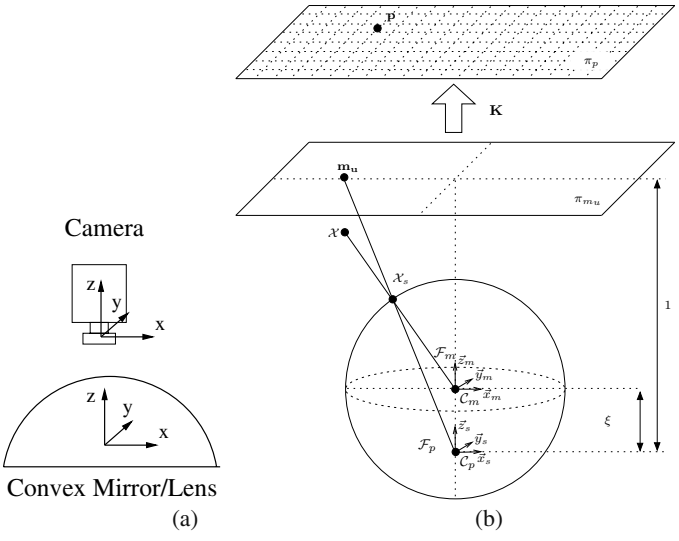


Fig. 13.1 (a) axis convention; and (b) unified image formation

where $\mathbf{L_s}$ is the interaction matrix related to $\mathbf{s}$. The control scheme is usually designed to reach an exponential decoupled decrease of the visual features to their desired value $\mathbf{s}^*$. If we consider an eye-in-hand system observing a static object, the control law is defined as

$$\mathbf{V_c} = -\lambda \widehat{\mathbf{L_s}}^{+} (\mathbf{s} - \mathbf{s}^*), \tag{13.4}$$

where $\widehat{\mathbf{L_s}}$ is a model or an approximation of $\mathbf{L_s}$, $\widehat{\mathbf{L_s}}^{+}$ the pseudo-inverse of $\widehat{\mathbf{L_s}}$, λ a positive gain tuning the time to convergence, and $\mathbf{V_c}$ the camera velocity sent to the low-level robot controller. The nonlinearities in system (13.4) explain the difference of behaviors in image space and in 3D space, and the inadequate robot trajectory that occurs sometimes when the displacement to realize is large (of course, for small displacements such that the variations of $\widehat{\mathbf{L_s}}$ are negligible, a correct behavior is obtained). An important issue is thus to determine visual features allowing to reduce the nonlinearities in (13.4). Furthermore, using (13.4) local minima can be reached when the number of features is not minimal. Therefore, one would like to chose a minimal representation (the number of features is equal to the number of DOF), but without singularities and robust with respect to image noise.

The problem of pose estimation consists in determining the rigid transformation ${}^c\mathbf{M}_o$ between the object frame $\mathcal{F}_o$ and the camera frame $\mathcal{F}_c$ in unknown position using the corresponding object image (see Figure 13.2). It is well known that the relation between an object point with coordinates $\mathcal{X}_c = (X_c, Y_c, Z_c, 1)$ in F_c and $\mathcal{X}_o = (X_o, Y_o, Z_o, 1)$ in F_o can be written as

$$\mathcal{X}_c = {}^c\mathbf{M}_o \mathcal{X}_o = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0_{31}} & 1 \end{bmatrix} \mathcal{X}_o. \tag{13.5}$$

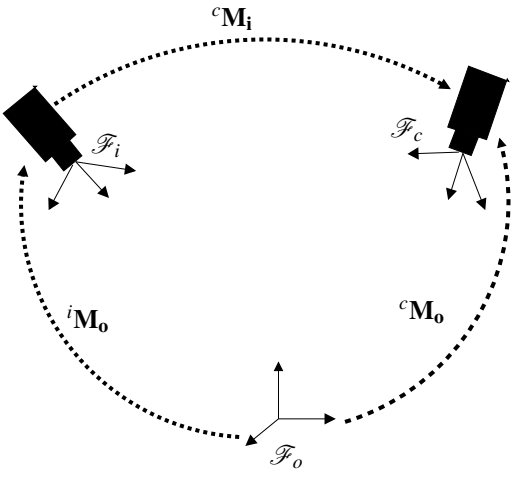


Fig. 13.2 Pose estimation using VVS

The matrix ${}^c\mathbf{M}_o$ can be estimated by minimizing the error module in image

$$e = \| \mathbf{s}({}^c\mathbf{M}_o) - \mathbf{s}^* \| \tag{13.6}$$

where $\mathbf{s}^*$ is the value of a set of visual features computed in the image acquired in the camera unknown position and $\mathbf{s}({}^\mathbf{c}\mathbf{M_o})$ is the value of the same set of features computed from the object model, the transformation ${}^c\mathbf{M}_o$, and the camera model. VVS consists in moving a virtual camera from a known initial pose ${}^i\mathbf{M}_o$ (referenced by the frame $\mathscr{F}_i$ on Figure 13.2) to the final unknown pose (referenced by the frame $\mathscr{F}_c$ on Figure 13.2) where e is minimized. In fact, the main difference between VS and VVS is that the visual features at each iteration are computed in VVS, while they are extracted from acquired images in VS. However, the displacement of the camera (real or virtual) is computed using the same control law (13.4).

13.3.2 Moments from the Projection of Points onto the Surface of Unit Sphere

In the following, the definition of the 3D moments computed from a discrete set of N points is firstly recalled. Then, the interaction matrix related to these moments computed from the point projection onto the unit sphere is given.

13.3.2.1 Definitions

The 3D moment of order $i+j+k$ computed from a discrete set of N points are defined by the classical equation

$$m_{i,j,k} = \sum_{h=1}^{N} x_{s_h}^i \, y_{s_h}^j \, z_{s_h}^k \tag{13.7}$$

where $(x_{s_h},\, y_{s_h},\, z_{s_h})$ are the coordinates of a 3D point. In our application, these coordinates are nothing but the coordinates of a point projected onto the unit sphere. They can be computed from the projection of a point onto the image plane and the inverse transform (13.2).

13.3.2.2 Interaction Matrix

In the case of moments computed from a discrete set of points, the derivative of (13.7) with respect to time is given by

$$\dot{m}_{i,j,k} = \sum_{h=0}^{N} (i\, x_{s_h}^{i-1}\, y_{s_h}^{j}\, z_{s_h}^{k} \dot{x}_{s_h} + j\, x_{s_h}^{i}\, y_{s_h}^{j-1}\, z_{s_h}^{k} \dot{y}_{s_h} + k\, x_{s_h}^{i}\, y_{s_h}^{j}\, z_{s_h}^{k-1} \dot{z}_{s_h}). \tag{13.8}$$

For any set of points (coplanar or noncoplanar), the interaction matrix related to $\mathbf{L}_{\mathbf{m_{i,j,k}}}$ can thus be obtained by combining (13.8) with the well known interaction matrix $\mathbf{L}_{\mathscr{X}_\mathbf{s}}$ of a point $\mathscr{X}_s$ on the unit sphere (defined such that $\dot{\mathscr{X}}_\mathbf{s} = \mathbf{L}_{\mathscr{X}_\mathbf{s}} \mathbf{V}$) [13, 28, 11]:

$$\mathbf{L}_{\mathscr{X}_\mathbf{s}} = \left[-\tfrac{1}{r}\mathbf{I_3} + \tfrac{1}{r}\mathscr{X}_s \mathscr{X}_s^\top \quad [\mathscr{X}_s]_\times \right] \tag{13.9}$$

where r is the distance of the 3D point to the sphere center. In the particular case of a coplanar set of points, the interaction matrix related to $m_{i,j,k}$ can be determined [28]:

$$\mathbf{L}_{m_{i,j,k}} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{wx} & m_{wy} & m_{wz} \end{bmatrix} \tag{13.10}$$

where

$$\begin{cases} m_{vx} = A(\beta_d m_{i+2,j,k} - i m_{i,j,k}) \\ \qquad + B(\beta_d m_{i+1,j+1,k} - i m_{i-1,j+1,k}) \\ \qquad + C(\beta_d m_{i+1,j,k+1} - i m_{i-1,j,k+1}), \\ m_{vy} = A(\beta_d m_{i+1,j+1,k} - j m_{i+1,j-1,k}) \\ \qquad + B(\beta_d m_{i,j+2,k} - j m_{i,j,k}) \\ \qquad + C(\beta_d m_{i,j+1,k+1} - j m_{i,j-1,k+1}) \\ m_{vz} = A(\beta_d m_{i+1,j,k+1} - k m_{i+1,j,k-1}) \\ \qquad + B(\beta_d m_{i,j+1,k+1} - k m_{i,j+1,k-1}) \\ \qquad + C(\beta_d m_{i,j,k+2} - k m_{i,j,k}), \\ m_{wx} = j m_{i,j-1,k+1} - k m_{i,j+1,k-1}\ , \\ m_{wy} = k m_{i+1,j,k-1} - i m_{i-1,j,k+1}\ , \\ m_{wz} = i m_{i-1,j+1,k} - j m_{i+1,j-1,k} \end{cases}$$

with $\beta_d = i + j + k$ and (A, B, C) are the parameters defining the object plane in the camera frame:

$$\frac{1}{r} = \boldsymbol{\alpha}^\top \mathscr{X}_s = A x_s + B y_s + C z_s. \tag{13.11}$$

13.4 Features Choice

In this section, the features choice is detailed. We will first explain how to obtain features to control the translational DOF with interaction matrices almost constant with respect to depth distributions. Then, a vector of features to control the whole 6 DOF will be proposed.

13.4.1 Invariants to Rotational Motion

The shape of an object does not change under rotational motions. After a rotational motion of the camera frame, it can easily be shown that the projected shape on the sphere also undergoes the same rotational motion. This means that the invariants to rotation in 3D space are also invariant if the considered points are projected onto the unit sphere. The decoupled control we propose is based on this invariance property. This important property will be used to select features invariant to rotations in order to control the 3 translational DOF. In this way, the following invariant polynomials to rotations have been proposed in [31] to control the translational DOF:

$$I_1 = m_{200}m_{020} + m_{200}m_{002} + m_{020}m_{002} - m_{110}^2 - m_{101}^2 - m_{011}^2, \tag{13.12}$$

$$\begin{aligned} I_2 = &- m_{300}m_{120} - m_{300}m_{102} + m_{210}^2 - m_{210}m_{030} - m_{210}m_{012} + m_{201}^2 - m_{201}m_{021} \\ &- m_{201}m_{003} + m_{120}^2 - m_{120}m_{102} + 3m_{111}^2 + m_{102}^2 - m_{030}m_{012} + m_{021}^2 - m_{021}m_{003} \\ &+ m_{012}^2, \end{aligned} \tag{13.13}$$

$$\begin{aligned} I_3 = &\, m_{300}^2 + 3m_{300}m_{120} + 3m_{300}m_{102} + 3m_{210}m_{030} + 3m_{210}m_{012} + 3m_{201}m_{021} \\ &+ 3m_{201}m_{003} + 3m_{120}m_{102} - 3m_{111}^2 + m_{030}^2 + 3m_{030}m_{012} + 3m_{021}m_{003} + m_{003}^2. \end{aligned} \tag{13.14}$$

The invariants (13.13) and (13.14) are of higher orders than (13.12). They are thus more sensitive to noise [23, 32]. For this reason, I_1 will be used in this chapter to control the translational DOF. Therefore, the set of points has to be separated in at least three subsets to get three independent values of I_1, which allows controlling the 3 translational DOF. Furthermore, in order to decrease the variations of the interaction with respect to depth distribution, it is $s_I = \frac{1}{\sqrt{I_1}}$ that will be used instead of I_1. This will be explained in the following.

13.4.2 Variation of the Interaction Matrix with respect to the Camera Pose

As it was mentioned above, one of the goals of this work is to decrease the non-linearity by selecting adequate features. In this way, the invariance property allows us to setup some interaction matrix entries to 0. These entries will thus be constant during the servoing task. However, the other entries depend on the camera pose as it

will be shown in the following. It will be also shown that the feature choice $s_I = \frac{1}{\sqrt{I_1}}$ allow obtaining interaction matrices almost constant with respect to the depth distribution.

13.4.2.1 Variation with respect to Rotational Motion

Let us consider two frames $\mathcal{F}_1$ and $\mathcal{F}_2$ related to the unit sphere with different orientations (${}^1\mathbf{R}_2$ is the rotation matrix between the two frames) but with the same center. In this case, the value of I_t is the same for the two frames, since it is invariant to rotational motions. Let $\mathcal{X}$ and $\mathcal{X}' = {}^2\mathbf{R}_1\mathcal{X}$ be the coordinates of a projected point in the frame $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively. Let us consider a function invariant to rotations $f(\mathcal{X}_1,\ldots,\mathcal{X}_N)$ that can be computed from the coordinates of N points onto the unit sphere (such as the invariants computed from the projection onto the unit sphere). The invariance condition between the frames $\mathcal{F}_1$ and $\mathcal{F}_2$ can thus be written as

$$f(\mathcal{X}'_1,\ldots,\mathcal{X}'_N) = f({}^2\mathbf{R}_1\mathcal{X}_1,\ldots,{}^2\mathbf{R}_1\mathcal{X}_N) = f(\mathcal{X}_1,\ldots,\mathcal{X}_N). \tag{13.15}$$

The interaction matrix that links the variation of the function f with respect to translational velocities can be obtained as

$$\mathbf{L}_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_{s_1}+\mathbf{T},\ldots,\mathcal{X}_N+\mathbf{T})}{\partial \mathbf{T}}, \tag{13.16}$$

where $\mathbf{T}$ is a small translational motion vector. Let us now apply this formula for the camera position defined by the frame $\mathcal{F}_2$:

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f}{\partial \mathbf{T}} = \frac{\partial f(\mathcal{X}'_1+\mathbf{T},\ldots,\mathcal{X}'_N+\mathbf{T})}{\partial \mathbf{T}} = \frac{\partial f({}^2\mathbf{R}_1\mathcal{X}_1+\mathbf{T},\ldots,{}^2\mathbf{R}_1\mathcal{X}_N+\mathbf{T}))}{\partial \mathbf{T}}. \tag{13.17}$$

From (13.17), it can be obtained that

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f\left({}^2\mathbf{R}_1(\mathcal{X}_1+{}^1\mathbf{R}_2\mathbf{T}),\ldots,{}^2\mathbf{R}_1(\mathcal{X}_N+{}^1\mathbf{R}_2\mathbf{T})\right)}{\partial \mathbf{T}}. \tag{13.18}$$

Combining this equation with the invariance to rotations condition (13.15), we get

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1+{}^1\mathbf{R}_2\mathbf{T},\ldots,\mathcal{X}_N+{}^1\mathbf{R}_2\mathbf{T})}{\partial \mathbf{T}}. \tag{13.19}$$

From this, we easily obtain

$$\mathbf{L}'_{\mathbf{f_v}} = \frac{\partial f(\mathcal{X}_1+\mathbf{T}',\ldots,\mathcal{X}_N+\mathbf{T}')}{\partial \mathbf{T}'}\frac{\partial \mathbf{T}'}{\partial \mathbf{T}} \tag{13.20}$$

where $\mathbf{T}' = {}^1\mathbf{R}_2\mathbf{T}$. Finally, combining (13.20) with (13.16), we obtain

$$\mathbf{L}'_{\mathbf{f_v}} = \mathbf{L}_{\mathbf{f_v}}\,{}^1\mathbf{R}_2. \tag{13.21}$$

This result was expected since applying a translational velocity $\mathbf{v_1}$ to the frame $\mathscr{F}_1$ is equivalent to applying a translational velocity to the frame $\mathscr{F}_2$ but taking into account the change of frame ($\mathbf{v_2} = {}^1\mathbf{R_2}\mathbf{v_1}$). This variation is thus natural, since the translational velocities to apply to the camera frame have to depend on its orientation. Finally, this result shows that rotational motions do not change the rank of the interaction matrix of the features used to control the translational DOF. In other words, the rotational motions do not introduce singularities on the interaction matrix and any rank change of the latter depends only on the translational motions.

13.4.2.2 Variation of the Interaction Matrix with respect to Depth

Obtaining constant interaction matrix entries means that the selected features depend linearly of the corresponding DOF. In this way, in [18, 30], it was shown that for good z-axis closed-loop behavior in IBVS, one should choose image features that scale as $s \sim Z$ (Z is the object depth). This means that the variation with respect to depth is a constant (*i.e.* the system is linear). In the case where the object is defined by an image region, the following feature has been proposed in [30] to control the motions along the optical axis:

$$s_r = \frac{1}{\sqrt{m_{00}}}$$

where m_{00} is the bidimensional moment of order 0 (that is the object surface in the image) using the conventional perspective projection model. In the case where the object is defined by a set of discrete points, the selected optimal feature was

$$s_d = \frac{1}{\sqrt{(\mu_{20} + \mu_{02})}} \tag{13.22}$$

where μ_{ij} are the central moments computed from a set of discrete points (see [30] for more details). Unfortunately, s_r and s_d allows only obtaining invariance to rotations around the optical axis and not to all 3D rotations. For this reason, $s_I = \frac{1}{\sqrt{I_1}}$ will be used instead of s_r. To explain the choice of $s_I = \frac{1}{\sqrt{I_1}}$, let us first determine how the polynomial invariant I_1 behaves when Z increases by considering each term of its formula. Let us consider the definition of the projection onto the unit sphere:

$$\begin{cases} x_s = \frac{X}{\sqrt{X^2+Y^2+Z^2}} \\ y_s = \frac{Y}{\sqrt{X^2+Y^2+Z^2}} \\ z_s = \frac{Z}{\sqrt{X^2+Y^2+Z^2}}. \end{cases} \tag{13.23}$$

From the definition of the projection onto the unit sphere, it can be obtained that if the depth Z increases (*i.e.* $X \ll Z$ and $Y \ll Z$), the point projection coordinates have the following behaviors with respect to depth: $x_s \sim \frac{1}{Z}$, $y_s \sim \frac{1}{Z}$ and $z_s \sim 1$. It follows

that: $m_{200} = \sum_{h=1}^{N} x_{s_h}^2 \sim \frac{1}{Z^2}$, $m_{020} = \sum_{h=1}^{N} y_{s_h}^2 \sim \frac{1}{Z^2}$, $m_{110} = \sum_{h=1}^{N} x_{s_h} y_{s_h} \sim \frac{1}{Z^2}$, $m_{101} = \sum_{h=1}^{N} x_{s_h} z_{s_h} \sim \frac{1}{Z}$, $m_{011} = \sum_{h=1}^{N} y_{s_h} z_{s_h} \sim \frac{1}{Z}$ and $m_{002} = \sum_{h=1}^{N} z_{s_h}^2 \sim N$. By neglecting the term depending on $\frac{1}{Z^4}$ when the depth increases enough, the polynomial can be approximated as

$$I_1 \approx N(m_{200} + m_{020}) - m_{100}^2 - m_{010}^2, \tag{13.24}$$

where N is the number of points. Therefore, it can be obtained that $I_1 \sim \frac{1}{Z^2}$ and $s_I = \frac{1}{\sqrt{I_1}} \sim Z$. Note that if the set of points is centered with respect to the optical axis (*i.e.* $m_{100} = m_{010} = 0$), we have

$$I_1 \approx N(m_{200} + m_{020}). \tag{13.25}$$

In this case, note the similarity between $s_I = \frac{1}{\sqrt{I_1}}$ and the features given by (13.22). In geometrical terms, if the set of points is centered with respect to the optical axis, the projection onto unit sphere and the projection onto a classical perspective behave in the same way when the depth increases. Besides, an example of interaction matrix variations with respect to depth distributions is given in Section 13.5.1.

13.4.3 Features Selection

We could consider the center of gravity of the object's projection onto the unit sphere to control the rotational DOF:

$$\mathbf{x_{s_g}} = \left(x_{s_g}, y_{s_g}, z_{s_g} \right) = \left(\frac{m_{100}}{m_{000}}, \frac{m_{010}}{m_{000}}, \frac{m_{001}}{m_{000}} \right).$$

However, only two coordinates of $\mathbf{x_{s_g}}$ are useful for the control since the point projection belongs to the unit sphere making one coordinate dependent of the others. That is why in order to control rotation around the optical axis, the mean orientation of all segments in the image is used as a feature. Each segment is built using two different points in an image obtained by re-projection to a conventional perspective plane.

Finally, as mentioned previously, the invariants to 3D rotation $s_I = \frac{1}{\sqrt{I_1}}$ are considered to control the translation. In practice, three separate set of points such that their centers are noncollinear can be enough to control the 3 translational DOF. In order to ensure the nonsingularity of the interaction matrix, the set of points is divided in four subsets (each subset has to encompass at least 3 points). This allows us to obtain four different features to control the 3 translational DOF.

13.5 Results

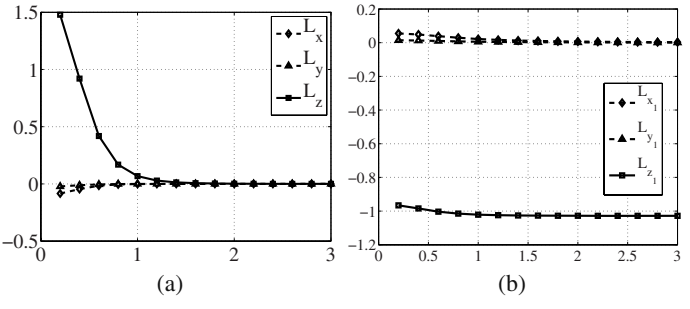
In this section, an example of interaction matrix variations with respect to depth distribution is given. Thereby, several results of pose estimation and visual servoing are presented.

13.5.1 Variation of the Interaction Matrix with respect to Depth Distribution

Figure 13.3 gives the variations of the interaction matrix entries of I_1 and $s_I = \frac{1}{\sqrt{I_1}}$ with respect to translational motion applied along the optical axis to the four random coplanar points defined in the camera frame as

$$\mathbf{X_o} = \begin{pmatrix} -0.3258 & -0.0811 & 0.1487 & 0.2583 \\ -0.0458 & 0.1470 & -0.1052 & 0.0039 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{pmatrix}. \tag{13.26}$$

The set of points has been chosen to be approximatively centered with respect to the $z-axis$ ($m_{100} \approx 0$ $m_{010} \approx 0$). For this reason, it can be seen that $L_x \approx L_{x_1} \approx L_y \approx L_{y_1} \approx 0$ ($\mathbf{L}_{I_1} = [L_x, L_y, L_z, 0, 0, 0]$ and $\mathbf{L}_{s_I} = [L_{x_1}, L_{y_1}, L_{z_1}, 0, 0, 0]$). In practice, the features I_1 and s_I also depend mainly on the translational motion with respect to the object axis of view. From Figures 13.3(a–b), it can be seen that L_{z_1} is almost constant and largely invariant to the object depth. On the other hand L_z decreases to 0 when the object depth increases.



(a) (b)

Fig. 13.3 Variations of the interaction matrix with respect to depth distribution (in meter): (a) results for I_1; and (b) results for $s_I = \frac{1}{\sqrt{I_1}}$

13.5.2 Visual Servoing Results

In these simulations, the set of points is composed of 4 noncoplanar points. For all the following simulations, the desired position corresponds to the 3D points coordinates defined in the camera frame as

$$\mathbf{X_d} = \begin{pmatrix} 0 & -0.2 & 0 & 0.2 \\ 0.2 & 0 & -0.2 & 0 \\ 0.9 & 1 & 1 & 1.2 \end{pmatrix}. \tag{13.27}$$

From the four set of points 4 different triangles can be obtained. For each triangle, the invariant $s_I = \frac{1}{\sqrt{I_1}}$ is computed to control the translational motion.

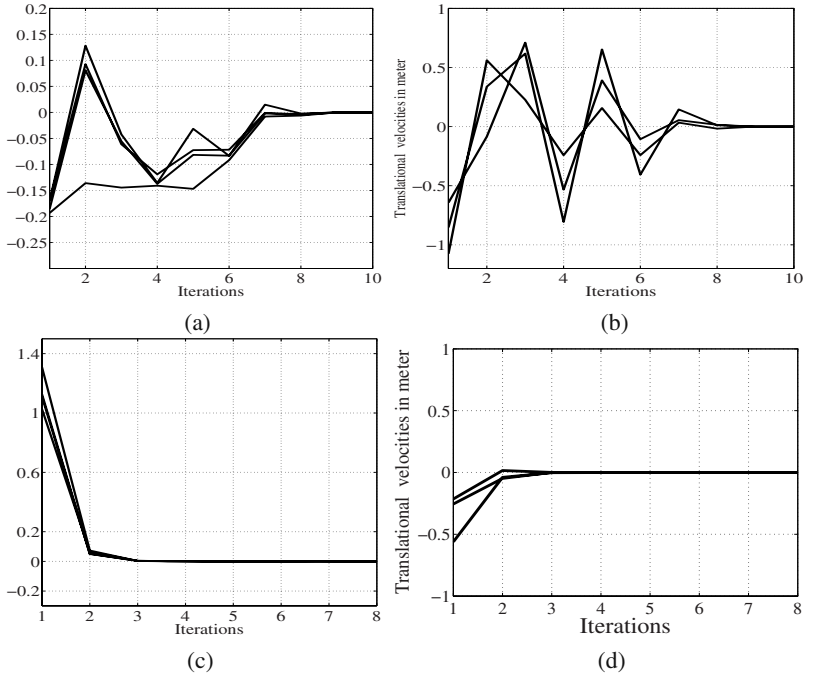


Fig. 13.4 Results using I_1: (a) features errors; (b) velocities (in m/s). Using $s_I = \frac{1}{\sqrt{I_1}}$: (c) features errors; and (d) velocities (in m/s)

In a first simulation, we show the advantage of using $s_I = \frac{1}{\sqrt{I_1}}$ instead of using directly I_1. For this purpose, the translational motion given by (13.28) has been considered between the desired and the initial camera poses. Further, in the control law (13.4), the scalar λ that tunes the velocity has been set to 1 and the interaction matrix computed at each iteration is used (*i.e.* $\widehat{\mathbf{L}}_\mathbf{s} = \mathbf{L}_\mathbf{s}$). If the system were completely linear, the convergence would be obtained in only one iteration. The nonlinearity of the system has as effect to damp or to magnify the camera velocities. In our case (*i.e.* $\lambda = 1$), the nonlinearity can slow the convergence (damping the velocity) or it can produce oscillations (magnifying the velocity). The results obtained using $I_t = \frac{1}{\sqrt{I_1}}$ and using I_1 are given on Figure 13.4. From Figures 13.4(a–b), oscillations can be observed for the features errors as well as for the velocities obtained using I_1 before converging (after 9 iterations). On the other hand, a fast convergence is obtained using $I_t = \frac{1}{\sqrt{I_1}}$ without oscillations (after only two iterations the system has almost converged). This shows that using $I_t = \frac{1}{\sqrt{I_1}}$, the system behaved almost as a linear system.

$$\mathbf{t_0} = \begin{pmatrix} 0.2, & 0.3, & 0.6 \end{pmatrix} \text{ m}. \tag{13.28}$$

In a second simulation, the rotational motion defined by the rotation vector (13.29) has been considered. The rotation matrix is obtained from the rotation vector $\theta\mathbf{u}$ using the well known Rodrigues formula. We compare the system behavior

using our features and using the point Cartesian coordinates (a conventional perspective projection has been considered). The obtained results are given on Figure 13.5. From Figure 13.5(a), it can be seen that a nice decrease of the features errors is obtained using our features. Furthermore, from Figure 13.5(b), since the considered translational motion is null, the translational velocity computed using the invariants to rotations are null (thanks to the invariance to rotations). Further, as for feature errors, Figure 13.5(c) shows a nice decrease of the rotational velocities. The results obtained using the point Cartesian coordinates to control the camera position are given on Figures 13.5(d–f). Figure 13.5(d) shows a nice decrease of the feature errors. On the other hand, the behavior of the velocities is far from satisfactory. Indeed, a strong translational motion is observed (see Figure 13.5(e)) and since the rotational DOF are coupled with the translational one, this introduced also a strong oscillations of the rotational velocities (see Figure 13.5(f)).

$$\boldsymbol{\theta}\mathbf{u} = \left(-6.42,\ 19.26,\ 128.40\right) \text{ deg.} \tag{13.29}$$

For the third simulation, the motion between the initial and desired camera poses defined by (13.30) and (13.31) has been considered. The same desired camera pose as for the first experiment was used. From Figure 13.6(a), it can be noticed that the feature errors behavior is very satisfactory. The same satisfactory behavior is obtained for translational and rotational velocities (see Figures 13.6(b–c)). Indeed, nice decreases of the feature errors as well as for the velocities are obtained. On the other hand the results obtained using the point Cartesian coordinates show a strong translational motion generated by the wide rotation and also oscillations of the whole velocities (see Figures 13.6(e–f))

$$\boldsymbol{\theta}\mathbf{u} = \left(-6.42,\ 19.26,\ 128.40\right) \text{ deg} \tag{13.30}$$

$$\mathbf{t_1} = \left(-0.,\ -0.3,\ 1\right) \text{ m.} \tag{13.31}$$

13.5.3 Pose Estimation Results

In this part, our pose estimation method is compared with the linear method proposed by Ansar in [1] and the iterative method proposed by Araujo [2]. The identity matrix has been used to initialize ${}^{i}\mathbf{M}_{o}$ for our method and for the iterative method proposed in [2]. The combination of the linear method and iterative method proposed by Araujo is also tested. In other words, the results obtained by the linear method will be used as initialization to the iterative method. The following setup has been used:

- an object composed of four points forming a square defined as follows has been considered:

$$X_o = \begin{bmatrix} -0.2 & 0.2 & -0.2 & 0.2 \\ -0.2 & -0.2 & 0.2 & 0.2 \\ 1 & 1 & 1 & 1 \end{bmatrix};$$

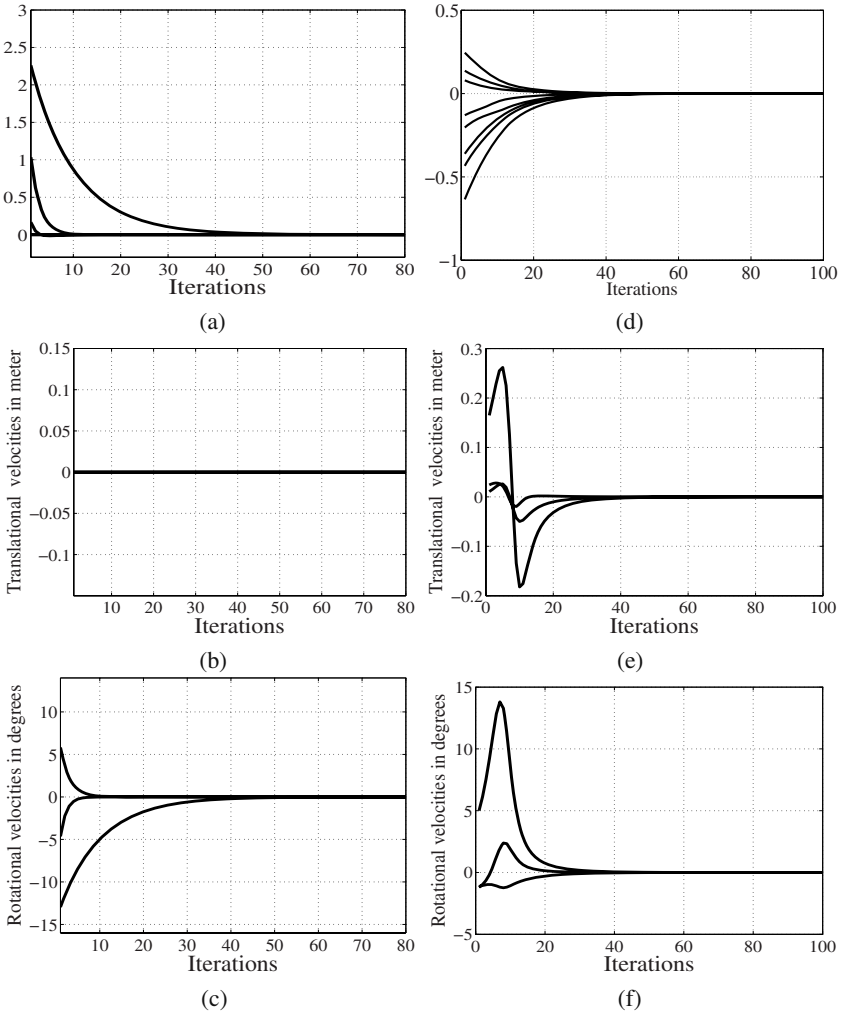


Fig. 13.5 Results for pure rotational motion. Using $S_I = \frac{1}{\sqrt{I_1}}$: (a) features errors; (b) translational velocities (in m/s); (c) rotational velocities (in deg/s). Using point coordinates: (d) features errors; (e) Translational velocities (in m/s); and (f) rotational velocities (in deg/s)

- a camera model with focal $F = 800$ and principal point coordinates $u = v = 400$ pixels has been used to compute the points coordinates in image;
- the interaction matrix corresponding to the current position is used in the control law (13.4) to compute the camera displacement (*i.e.* $\widehat{\mathbf{L}}_\mathbf{s} = \mathbf{L}_\mathbf{s}$); and
- random poses have been generated as follow:
  - 1000 random translational motions $\mathbf{t} = (1\sigma_1 \;\; 1\sigma_2 \;\; 1.3\sigma_3)$ are firstly applied to the point coordinates defined in the square frame, where σ_1 and σ_2 are

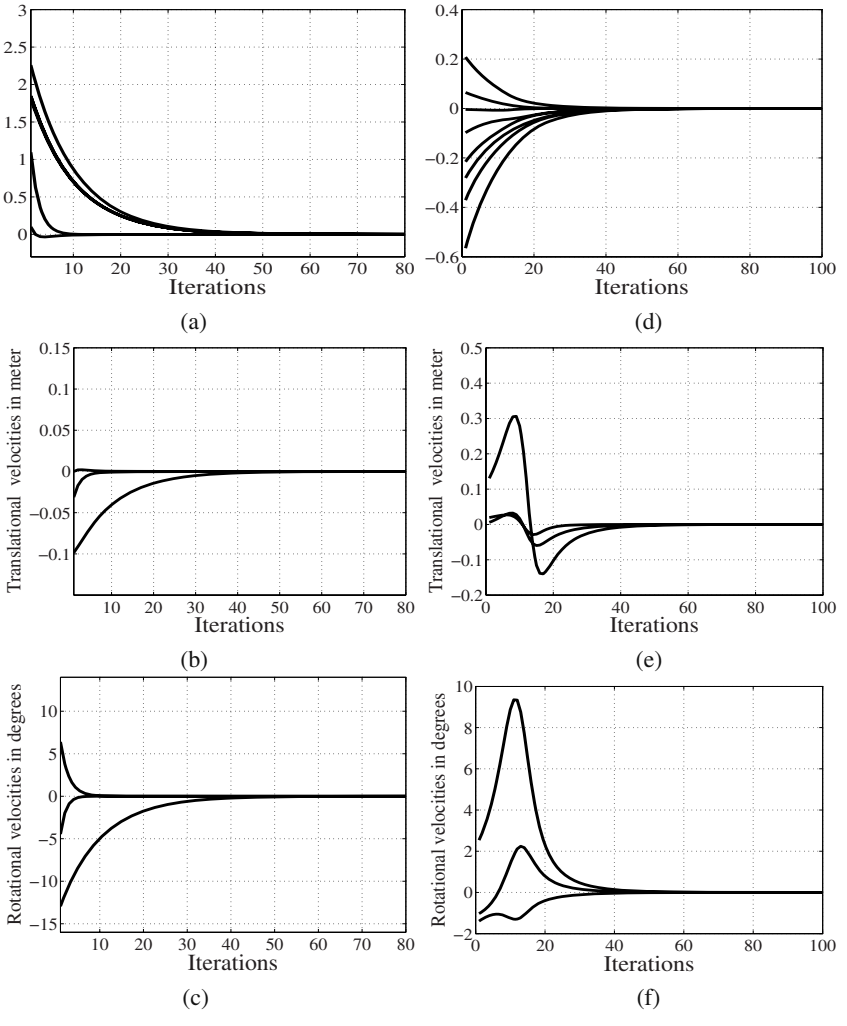


Fig. 13.6 Results for general motion. Using $I_t = \frac{1}{\sqrt{I_1}}$: (a) features errors; (b) translational velocities (in m/s); (c) rotational velocities (in deg/s). Using point coordinates: (d) features errors; (e) translational velocities (in m/s); and (f) rotational velocities (in deg/s)

random numbers chosen from a normal distribution with mean zero, variance one and standard deviation one, σ_3 is a random number chosen from a uniform distribution on the interval $[0.0\ 1.0]$; and

– the rotational motion is chosen such that the points coordinates belongs to the image limits $[1\ \ 800; 1\ \ 800]$. Further, the rotational motion with respect to the optical axis can range randomly between $[0\ \ 2\pi]$.

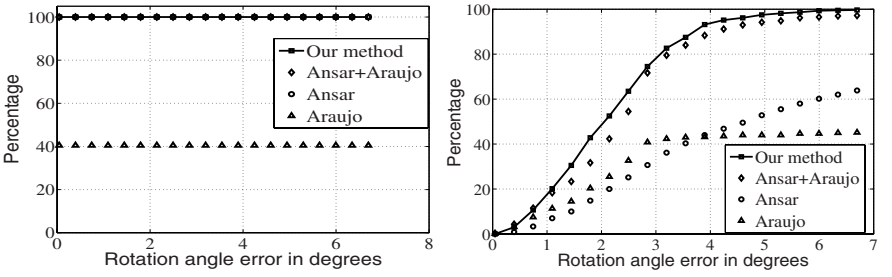


Fig. 13.7 Distribution of θ_e with respect to noise level and errors on camera parameters: (a) results with zero noise and exact camera; and (b) results for 0.3 pixels gaussian noise, principal points coordinates [375 375] and focal $F = 760$

Let us consider the pose error defined by

$$\mathbf{T_e} = \begin{pmatrix} \mathbf{R_e} & \mathbf{t_e} \\ \mathbf{0_{1\times 3}} & 1 \end{pmatrix} = \mathbf{T_r^{-1}T_c}$$

where $\mathbf{T_r}$ and $\mathbf{T_c}$ are respectively the real and the estimated pose. If the correct pose is obtained, $\mathbf{T_e}$ is equal to the identity matrix ($\| \mathbf{t_e} \| = 0$ and $\mathbf{R_e} = \mathbf{I_3}$). Let θ_e be the rotation error corresponding to the rotation matrix $\mathbf{R_e}$. Figures 13.7 and 13.8 give the distribution of θ_e and $\| \mathbf{t_e} \|$ using the four different methods and for three different levels of noise. In other words, for each values of θ_e and $\| \mathbf{t_e} \|$, the plot gives the percentage of the errors smaller or equal to these values.

Figures 13.7(a) and 13.8(a) give the distributions of θ_e and $\| \mathbf{t_e} \|$ when perfect data is considered (without noise and using the exact camera parameters). From these figures, it can be seen that the linear method, our method and **AA** method have always estimated the exact pose. On the other hand, Araujo's method initialized by the identity matrix only converges for nearly 40% cases. Figures 13.7(b) and

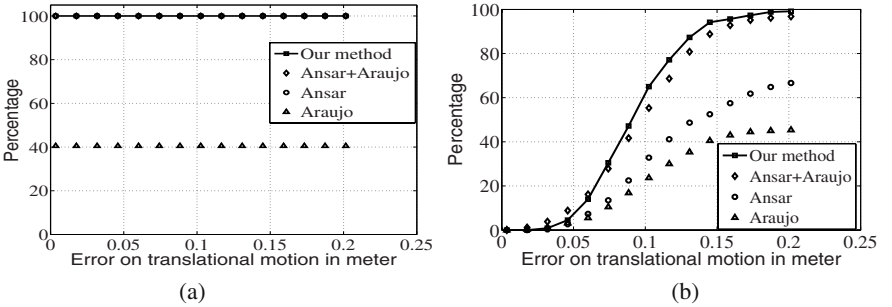


Fig. 13.8 Distribution of $\| \mathbf{t_e} \|$ with respect to noise level and errors on camera parameters: (a) results with zero noise; and (b) results for 0.3 pixels gaussian noise, principal points coordinates [375 375] and focal $F = 760$

13.8(b) give the obtained results with 0.3 pixels gaussian noises, principal points coordinates $[375\ \ 375]$ and focal $F = 760$ (recall that the real values are $[400\ \ 400]$ for the principal point and $F = 800$ for the focal). From these figures, it can be noticed that the accuracy of the estimation using the linear method decreases when the data noise increases. The results obtained using the linear method are improved using Araujo's method. On the other hand, the accuracy of the Araujo iterative method initialized by the identity matrix also decreases, but the convergence percentage is still around 40%. Finally for the same experiment, our iterative method converges for all cases and gives more accurate estimation of the poses.

13.6 Conclusion

In this chapter, a unique and efficient decoupled scheme for visual servoing and pose estimation has been proposed. The proposed scheme is valid for cameras obeying the unified model. More precisely, the invariants to rotational motions computed from the projection onto the unit sphere are used to control the translational DOF. Adequate forms of invariants have been proposed to decrease the interaction matrix variations with respect to the depth distributions.The validations results have shown the efficiency of the proposed scheme. Future works will be devoted to extend these results to model-free pose estimation problem and to region-based visual servoing.

References

[1] Ansar, A., Daniilidis, K.: Linear pose estimation from points or lines. IEEE Trans. on Pattern Analysis and Machine Intelligence 25, 282–296 (2003)
[2] Araujo, H., Carceroni, R.L., Brown, C.M.: A fully projective formulation to improve the accuracy of lowe's pose-estimation algorithm. Computer Vision and Image Understanding 70, 227–238 (1998)
[3] Baker, S., Nayar, S.: A theory of catadioptric image formation. Int. Journal of Computer Vision 35(2), 175–196 (1999)
[4] Barreto, J., Araujo, H.: Issues on the geometry of central catadioptric image formation. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 2, pp. II-422–II-427 (2001)
[5] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman, D., Hager, G., Morse, A. (eds.) The Confluence of Vision and Control. LNCIS, vol. 237, pp. 66–78. Springer, Heidelberg (1998)
[6] Corke, P.I., Hutchinson, S.A.: A new partitioned approach to image-based visual servo control. IEEE Trans. on Robotics and Automation 17(4), 507–515 (2001)
[7] Dementhon, D., Davis, L.: Model-based object pose in 25 lines of code. Int. Journal of Computer Vision 15(1-2), 123–141 (1995)
[8] Dhome, M., Richetin, M., Lapreste, J.T., Rives, G.: Determination of the attitude of 3d objects from a single perspective view. IEEE Trans. on Pattern Analysis and Machine Intelligence 11(12), 1265–1278 (1989)
[9] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Trans. on Robotics and Automation 8, 313–326 (1992)

[10] Fiore, P.: Efficient linear solution of exterior orientation. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(2), 140–148 (2001)
[11] Fomena, R., Chaumette, F.: Visual servoing from spheres using a spherical projection model. In: IEEE International Conference on Robotics and Automation, 2007, pp. 2080–2085 (2007)
[12] Geyer, C., Daniilidis, K.: Mirrors in motion: Epipolar geometry and motion estimation. Int. Journal on Computer Vision 45(3), 766–773 (2003)
[13] Hamel, T., Mahony, R.: Visual servoing of an under-actuated dynamic rigid body system: an image-based approach. IEEE Trans. on Robotics and Automation 18(2), 187–198 (2002)
[14] Iwatsuki, M., Okiyama, N.: A new formulation of visual servoing based on cylindrical coordinates system with shiftable origin. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Lausanne, Switzerland, pp. 354–359 (2002)
[15] Lee, J.S., Suh, I., You, B.J., Oh, S.R.: A novel visual servoing approach involving disturbance observer. In: IEEE Int. Conf. on Robotics and Automation, ICRA 1999, Detroit, Michigan, pp. 269–274 (1999)
[16] Lowe, D.G.: Fitting parameterized three-dimensional models to images. IEEE Trans. on Pattern Analysis and Machine Intelligence 13, 441–450 (1991)
[17] Lu, C.P., Hager, G., Mjolsness, E.: Fast and globally convergent pose estimation from video images. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(6), 610–622 (2000)
[18] Mahony, R., Corke, P., Chaumette, F.: Choice of image features for depth-axis control in image-based visual servo control. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2002, Lausanne, Switzerland, vol. 1, pp. 390–395 (2002)
[19] Malis, E., Chaumette, F., Boudet, S.: 2 1/2 d visual servoing. IEEE Trans. on Robotics and Automation 15(2), 238–250 (1999)
[20] Marchand, E., Chaumette, F.: Virtual visual servoing: A framework for real-time augmented reality. In: Drettakis, G., Seidel, H.P. (eds.) EUROGRAPHICS 2002 Conference Proceeding, Computer Graphics Forum, Saarebrücken, Germany, vol. 21(3), pp. 289–298 (2002)
[21] Martinet, P., Gallice, J.: Position based visual servoing using a nonlinear approach. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 1999, Kyongju, Korea, vol. 1, pp. 531–536 (1999)
[22] Mei, C., Rives, P.: Single view point omnidirectional camera calibration from planar grids. In: 2007 IEEE International Conference on Robotics and Automation, pp. 3945–3950 (2007)
[23] Mukundan, R., Ramakrishnan, K.R.: Moment Functions in Image Analysis Theory and Application. M. World Scientific Publishing, Co. Pte. Ltd., Singapore (1998)
[24] Rives, P., Azinheira, J.: Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2004, New Orleans, Louisiana, pp. 255–260 (2004)
[25] Safaee-Rad, R., Tchoukanov, I., Smith, K., Benhabib, B.: Three-dimensional location estimation of circular features for machine vision. IEEE Trans. onRobotics and Automation 8(5), 624–640 (1992)
[26] Sundareswaran, V., Behringer, R.: Visual servoing-based augmented reality. In: IWAR 1998: Proceedings of the international workshop on Augmented reality: placing artificial objects in real scenes, pp. 193–200. A. K. Peters, Ltd., Natick (1999)
[27] Svoboda, T., Pajdla, T.: Epipolar geometry for central catadioptric cameras. Int. Journal on Computer Vision 49(1), 23–37 (2002)

[28] Tahri, O.: Utilisation des moments en asservissement visuel et en calcul de pose. Ph.D. thesis, University of Rennes (2004)
[29] Tahri, O., Chaumette, F.: Complex objects pose estimation based on image moment invariants. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2005, Barcelona, Spain, pp. 438–443 (2005)
[30] Tahri, O., Chaumette, F.: Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics 21(6), 1116–1127 (2005)
[31] Tahri, O., Chaumette, F., Mezouar, Y.: New decoupled visual servoing scheme based on invariants from projection onto a sphere. In: IEEE Int. Conf. on Robotics and Automation, ICRA 2008, pp. 3238–3243 (2008)
[32] Teh, C., Chin, R.T.: On image analysis by the method of moments. IEEE Trans. on Pattern Analysis and Machine Intelligence 10(4), 496–513 (1988)
[33] Wilson, W., Hulls, C., Bell, G.: Relative end-effector control using cartesian position-based visual servoing. IEEE Trans. on Robotics and Automation 12(5), 684–696 (1996)

Chapter 14
Gradient Projection Methods for Constrained Image-based Visual Servo

Wenzhuo Ma and Seth Hutchinson

Abstract. This chapter describes redundancy-based solutions to visual servo problems that require honoring constraints on camera motion that are imposed either by the environment or by physical limits on the robot system. The task function approach provides a formalism by which a secondary task can be achieved without disturbing the execution of the primary task. For problems considered here, secondary tasks correspond to enforcing various constraints. With gradient projection methods, the secondary task corresponds to the gradient of a cost function, and this gradient is projected onto the null space of the primary task to yield motions that will honor the constraints without interfering with the performance of the primary task. The chapter reprises the development of gradient projection methods in the context of constrained visual servo control, and presents simulation results to illustrate performance of the approach.

14.1 Introduction

In this chapter we consider the problem of performing visual servo tasks while respecting constraints on camera motion. In particular, we consider constraints on camera motion that derive from robot joint limits, the possibility of occlusion, and the need to keep objects of interest in the field of view during task execution. While there are many scenarios in which this type of constrained visual servo would be appropriate, we limit our attention here to the case in which the visual servo system is redundant with respect to the task (*i.e.*, the camera motion has more degrees of freedom (DOF) than are required to execute the task). In such cases, constraints can be formulated as a secondary task, and the redundant DOF can be used to satisfy those constraints [15, 12].

Wenzhou Ma and Seth Hutchinson
Department of Electrical and Computer Engineering, University of Illinois,
405 North Matthews Avenue, Urbana, IL 61801, USA
e-mail: seth@illinois.edu

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 253–274.
springerlink.com

The task function approach [14] provides a formalism by which the motion needed to accomplish the secondary task can projected onto the null space of the primary task, yielding a total motion that satisfies the primary task while making progress toward the secondary task. Here, we use the method of gradient projection [13], and encode constraints using an objective function whose gradient is projected onto the null space of the primary task. While better methods exist for individual, specific problems, the purpose of this chapter is to give a unified treatment of gradient projection methods as they can be used to solve a variety of constrained visual servo problems.

The methods described below have been developed in the context of redundant robots [15, 16, 12] and visual servo control [8, 4, 10, 9]. Our development follows that given in these references. The chapter is organized as follows. In Section 14.2, we review null space methods, and in particular the gradient projection method, for controlling the motion of redundant robots. Following this, we describe how gradient projection methods can be used to avoid joint limits (Section 14.3), prevent occlusion (Section 14.4), and keep an object of interest in the field of view (Section 14.5).

14.2 Exploiting Redundancy by Projecting onto Null Spaces

A system is said to have redundancy if it has more DOF than are necessary to perform a given task. Since 6 DOF are enough for full position and orientation control of an end effector, a manipulator with 7 DOF can be said to be redundant. Systems with 6 or fewer DOF are regarded to have redundancy if the particular task to be performed requires fewer than 6 DOF.

In this section, we describe the mathematical formalism by which redundancy can be exploited to make progress toward secondary goals while performing a primary task. In particular, we describe below task decomposition and gradient projection methods [12, 15, 5].

14.2.1 Task-decomposition Approach

Most complicated tasks given to a manipulator can be formulated in such a way that the task is broken down into several subtasks with a priority order. Each subtask is performed using the DOF that remain after all the subtasks with higher priority have been implemented. Control problems for redundant manipulators can be approached via the subtask method by regarding a task to be performed by a redundant manipulator as the task with highest priority.

Consider a manipulator with n DOF. The joint variable of the ith joint is $q_i, (i = 2, 3, ..., n)$. The manipulator configuration is denoted by the vector $\boldsymbol{q} = [q_1, q_2, ..., q_n]$. Assume that the first subtask can be described properly by an m_1-dimensional vector, y_1, which is a function of $\boldsymbol{q}$:

$$y_1 = f_1(q). \tag{14.1}$$

Assume also that the desired trajectory for y_1 is given by $y_{1d}(t)$. A vector which is suitable for describing a manipulation task is called a manipulation vector.

Subsequent subtasks of lower priority could be specified in a variety of ways. The two most common approaches are as follows:

1. an m_2-dimensional manipulation vector y_2 is given by $y_2 = f_2(q)$ and the second subtask is specified by the desired trajectory $y_{2d}(t)$;
2. a criterion function $p = V(q)$ is given, and the second subtask is to keep this criterion as large as possible.

In either of these cases, the typical way to address lower-priority tasks is to exploit the null space associated to the primary task. Differentiating (14.1) with respect to time yields

$$\dot{y}_1 = J_1 \dot{q}, \tag{14.2}$$

where J_1 is the Jacobian matrix of y_1 respect to q. When the desired trajectory y_{1d} is given, the general solution for $\dot{q}$ is

$$\dot{q} = J_1^+ \dot{y}_{1d} + (I - J_1^+ J_1) k_1, \tag{14.3}$$

where k_1 is an n-dimensional arbitrary constant vector. The first term on the right-hand side is the joint velocity to achieve the desired trajectory, $y_{1d}(t)$. When there are multiple solutions for $\dot{q}$ satisfying the equation, this term gives a solution that minimizes $\|\dot{q}\|$, the Euclidean norm of $\dot{q}$. The second term on the right-hand side reflects the redundancy remaining after performing the first subtask.

Consider the case when the second subtask is specified by the desired trajectory $y_{2d}(t)$ of the manipulation variable y_2. Differentiating $y_2 = f_2(q)$, we have

$$\dot{y}_2 = J_2 \dot{q}. \tag{14.4}$$

Substituting $\dot{y}_2 = \dot{y}_{2d}$ and (14.3) into (14.4), we obtain

$$\dot{y}_{2d} - J_2 J_1^+ \dot{y}_{1d} = J_2 (I - J_1^+ J_1) k_1. \tag{14.5}$$

For the linear $Ax = b$, the general solution is given by

$$x = A^+ b + (I - A^+ A) k, \tag{14.6}$$

where A^+ is the pseudo-inverse of A. Letting $\tilde{J}_2 = J_2 (I - J_1^+ J_1)$, from (14.5) and (14.6), we obtain

$$k_1 = \tilde{J}_2^+ (\dot{y}_{2d} - J_2^+ J_1 \dot{y}_{1d}) + (I - \tilde{J}_2^+ \tilde{J}_2) k_2, \tag{14.7}$$

where k_2 is an n-dimensional arbitrary constant vector. Note that the relation

$$(I - J_1^+ J_1) \tilde{J}_2^+ = \tilde{J}_2^+ \tag{14.8}$$

holds. Therefore, substituting (14.7), (14.8) into (14.3) we obtain

$$\dot{q}_d = J_1^+ \dot{y}_{1d} + \tilde{J}_2^+ (\dot{y}_{2d} - J_2^+ J_1 \dot{y}_{1d}) + (I - J_1^+ J_1 - \tilde{J}_2^+ \tilde{J}_2) k_2 \tag{14.9}$$

as the desired joint velocity that first realizes trajectory y_{1d} and then realizes y_{2d} as closely as possible using the remaining redundancy. This approach iteratively constructs the equations used to perform more subtasks, in this case using k_2 to perform the third task, if $I - J_1^+ J_1 - \tilde{J}_2^+ \tilde{J}_2$ is not zero.

In the case when the second subtask is specified by a criterion function, we select the vector k_1 to make the criterion p as large as possible. One natural approach is to determine k_1 by the following equations:

$$k_1 = \kappa g \tag{14.10}$$

$$g = \nabla_q V, \tag{14.11}$$

where $g \in R^n$, and κ is an appropriate positive constant. The desired joint velocity, $\dot{q}_d$, is given by

$$\dot{q}_d = J_1^+ \dot{y}_{1d} + \kappa (I - J_1^+ J_1) g. \tag{14.12}$$

Since p is the criterion function, from (14.12), we have

$$\dot{p} = \dot{V}(q) = \frac{\partial V}{\partial q} \dot{q} = g^T J_1^+ \dot{y}_{1d} + \kappa g^T (I - J_1^+ J_1) g. \tag{14.13}$$

Since $(I - J_1^+ J_1)$ is nonnegative definite, the second term on the right-hand side of (14.13) is always nonnegative, causing the value of criterion p to increase.

With this approach, k_1 is the gradient of the function $V(q)$ and the vector $\kappa(I - J_1^+ J_1)g$ corresponds to the orthogonal projection of k_1 on the null space of J_1. The constant κ is chosen so as to make p increase as quickly as possible under the condition that $\dot{q}_d$ does not become excessively large.

14.2.2 The Gradient Projection Method Applied to Robotic Control

The optimization problem is typically represented by a constraint set X and a cost function f that maps elements of X into real numbers [13, 1]. A vector $x^* \in X$ is a local minimum of f over the set X if it is no worse than its feasible neighbors; that is, if there exists an $\varepsilon > 0$ such that

$$f(x^*) \leq f(x), \forall x \in X \quad with \parallel x - x^* \parallel < \varepsilon. \tag{14.14}$$

A vector $x^* \in X$ is a global minimum of f over the set X if it is no worse than all other feasible vectors, that is,

$$f(x^*) \leq f(x), \forall x \in X. \tag{14.15}$$

The local or global minimum x^* is said to be strict if the corresponding inequality above is strict for $x \neq x^*$.

Gradient descent is a popular optimization method that finds a local minimum of a function by successively taking steps proportional to the negative of the gradient of the cost function. Gradient descent is also known as steepest descent, or the method of steepest descent. Gradient descent is based on the observation that if the real-valued function $f(x)$ is defined and differentiable in a neighborhood of a point a, then $f(x)$ decreases fastest if one goes from a in the direction of the negative gradient of f at a, it follows that, if $b = a - \alpha\nabla f(a)$ for $\alpha > 0$ a small enough number, then $f(a) \geq f(b)$. The algorithm starts with a guess x_0 for a local minimum of f, and constructs the sequence $x_0, x_1, x_2,...$ such that

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), k \geq 0. \tag{14.16}$$

We have

$$f(x_0) \geq f(x_1) \geq f(x_2) \geq ... \tag{14.17}$$

According to the Armijo rule, the sequence converges to the desired local minimum if the cost improvement obtained at each iteration is sufficiently large. The value of the step size α is allowed to change at every iteration.

For constrained optimization problems, gradient descent may not yield a feasible point at each iteration (*i.e.*, x_{k+1} is not guaranteed to satisfy the constraints). In this case, the gradient projection method can be used to ensure that each x_k is feasible. The simplest gradient projection method determines a feasible direction method of the form

$$x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k), \tag{14.18}$$

where

$$\bar{x}_k = [x_k - s_k \nabla f(x_k)]^+ . \tag{14.19}$$

Here, $[.]^+$ denotes projection on the set X, $\alpha_k \in (0,1]$ is a step size, and s_k is a positive scalar. Thus, to obtain the vector $\bar{x}_k$, we take a step $-s_k\nabla f(x_k)$ along the negative gradient, as in steepest descent. We then project the result $x_k - s_k\nabla f(x_k)$ on X, thereby obtaining the feasible vector $\bar{x}_k$. Finally, we take a step along the feasible direction $\bar{x}_k - x_k$ using the step size α_k.

The gradient projection method can be applied to the control of redundant robots (see, *e.g.*, [15]). In this case, constraints on robot motion are described by a cost function. The gradient of the cost function can be considered as an artificial force, pushing the robot away from the undesirable configurations. At each iteration, an artificial force $g(q)$ is induced by the cost function at the current position. The gradient of this function, projected onto the null space of the main task Jacobian, is used to produce the motion necessary to minimize the specified cost function as far as possible.

The main advantage of this method is that the constraint avoidance process has no effect on the main task. This can be understood as follows [12]. A square matrix M is called an orthogonal projection if its mapping Mx of x is perpendicular to $x - Mx$ for any x. It is noteworthy that $JJ^+, J^+J, I - JJ^+, I - J^+J$ are all orthogonal projections. We denote the range space of J^+ by $\mathscr{R}(J^+)$, and the range space of $I - J^+J$ by $\mathscr{R}(I - J^+J)$. We denote the null space of J by $\mathscr{N}(J)$, and the orthogonal complements of

$\mathscr{N}(J)$ by $\mathscr{N}(J)^{\perp}$, respectively. These have the following relationships with the pseudo-inverse of J.

$$\mathscr{N}(J) = \mathscr{R}(I - J^{+}J) \tag{14.20}$$

$$\mathscr{N}(J)^{\perp} = \mathscr{R}(J^{+}). \tag{14.21}$$

The range of J^{+} and the range of $I - J^{+}J$ are orthogonal, and their projections are in different spaces. Therefore, the implementation of the second task does not disturb the fist manipulation variable.

Let us consider the problem:

$$\min V(q), \quad q \in \mathbb{R}^{k}, \tag{14.22}$$

where k is the number of robot joints. The classical solution is to move the robot according to the gradient of the cost function, computed in the articular space

$$\dot{q} = \kappa g(q) = -\kappa Q \nabla_{q} V, \tag{14.23}$$

where κ is a positive scalar, used as a gain, and Q is a constant positive matrix. Pre-multiplying (14.23) by $\nabla_{q}^{T} V$, we get

$$\frac{d}{dt} V(q) = -\kappa \nabla_{q}^{T} V Q \nabla_{q} V \leq 0. \tag{14.24}$$

Thus, V decreases with time as long as $\nabla_{q} V \neq 0$, and remains constant when $\nabla_{q} V = 0$. It is not uncommon to choose Q as the identity matrix I.

Consider now a potential field $V_{\Phi} = V(\Phi(q))$. Using the chain rule, we have

$$\dot{\Phi} = \left(\frac{\partial \Phi}{\partial q} \right)^{T} \dot{q},$$

and combining this with (14.23) we obtain

$$\dot{\Phi} = \frac{\partial \Phi}{\partial q} \dot{q} = -\kappa \left(\frac{\partial \Phi}{\partial q} \right)^{T} Q \nabla_{q} V = -\kappa \left(\frac{\partial \Phi}{\partial q} \right)^{T} Q \left(\frac{\partial \Phi}{\partial q} \right) \nabla_{\Phi} V. \tag{14.25}$$

If we choose

$$Q = \left(\frac{\partial \Phi}{\partial q} \right)^{\dagger} \left(\frac{\partial \Phi}{\partial q} \right)^{+}, \tag{14.26}$$

in which the superscript † denotes the transpose of the pseudo-inverse, (14.24) is verified because Q is a positive matrix. Substituting (14.26) into (14.25), we obtain

$$\dot{\Phi} = -\kappa \nabla_{\Phi} V. \tag{14.27}$$

From (14.23) and (14.26), we obtain

$$\dot{q} = -\left(\frac{\partial \Phi}{\partial q} \right)^{\dagger} \nabla_{\Phi} V_{\Phi}, \tag{14.28}$$

where we can note the use of the Jacobian pseudo-inverse. Classical methods propose generally to use simply the transpose of the Jacobian, leading to $\dot{q} = -(\frac{\partial \Phi}{\partial q})^T \nabla_\Phi V_\Phi$. Since the pseudo-inverse provides the least-square solution, the resulting artificial force is the most efficient one at equivalent norm.

Considering now several minimization problems $V^i = V^i_{\Phi_i}$, where Φ_i are different parameterizations. The global cost function can be written:

$$V = \sum_i \gamma_i V^i_{\Phi_i}, \tag{14.29}$$

where the scale factors γ_i are used to adjust the relative influence of the different forces. The force realizing a trade-off between these constraints is thus:

$$g = \sum_i \gamma_i g^i_{\Phi_i} = \sum_i \gamma_i \left(\frac{\partial \Phi_i}{\partial q} \right)^\dagger \nabla_{\Phi_i} V^i_{\Phi_i}. \tag{14.30}$$

As to the final control law, the gradient g is used as the second task. It has thus to be projected onto the null space of the first task. Substituting $q_1 = J_1^+ \dot{y}_{1d}$ and $P_1 = (I - J_1^+ J_1)$ into (14.12), the complete control law is finally

$$\dot{q}_d = \dot{q}_1 - \kappa P_1 g. \tag{14.31}$$

The choice of the parameter κ is very important. If κ is too small, the gradient force may be too small to respect the constraints. Besides, if κ is too high, some overshoot can occur in the computed velocity.

14.3 Gradient Projection Method for Avoiding Joint Limits

In this section we describe how the gradient projection method can be used to avoid reaching joint limits while performing visual servo tasks. The mathematical development below is based on the work reported in [4]. We present a set of simulation results that illustrate the performance of the method.

14.3.1 Basic Algorithm Design

The cost function proposed in [4] for joint-limit avoidance is defined directly in the configuration (joint) space. It reaches its maximal value at the joint limits, and is constant far from the limits, so that the gradient is zero. The robot lower and upper joint limits for each axis i are denoted $q_i^{\min}$ and $q_i^{\max}$, and the robot configuration q is said to be admissible if, for all i, we have $q_i^{\min} \leq q_i \leq q_i^{\max}$.

Activation thresholds, $q_{li}^{\min}$ and $q_{li}^{\max}$ are defined in terms of the joint limits so that when the joint value q_i crosses an activation threshold the cost function begins to grow rapidly. These activation levels can be defined in terms of the joint limits, the interval $\bar{q}_i = q_i^{\max} - q_i^{\min}$, and the fraction of the interval over which the cost function should remain nearly constant

$$q_{li}^{\min} = q_i^{\min} + \rho\bar{q} \tag{14.32}$$

$$q_{li}^{\max} = q_i^{\max} - \rho\bar{q}, \tag{14.33}$$

in which $0 < \rho < 0.5$ is a tuning parameter (in [4], the value $\rho = 0.1$ is typically used).

The cost function used to avoid joint limits V^{jl} is given by

$$V^{jl}(q) = \frac{1}{2}\sum_{i=1}^{n}\frac{\delta_i^2}{\bar{q}_i}, \tag{14.34}$$

where

$$\delta_i = \begin{cases} q_i - q_{l_i}^{\min}, & \text{if } q_i < q_{l_i}^{\min} \\ q_i - q_{l_i}^{\max}, & \text{if } q_i > q_{l_i}^{\max} \\ 0, & \text{else} \end{cases}. \tag{14.35}$$

The gradient of the cost function is easily obtained as

$$g(q) = \begin{cases} \frac{q_i - q_{l_i}^{\max}}{\Delta\bar{q}_i}, & q_i > q_{l_i}^{\max} \\ \frac{q_i - q_{l_i}^{\min}}{\Delta\bar{q}_i}, & q_i < q_{l_i}^{\min} \\ 0, & \text{else} \end{cases}. \tag{14.36}$$

To apply this method in an image-based visual servoing (IBVS) approach [2, 3], it is necessary to relate changes in the image to joint velocities. To this end, IBVS control laws are typically expressed in the operational space (*i.e.*, in the camera frame), and then computed in the joint space using the robot inverse Jacobian. IBVS can be augmented to include the avoidance of joint limits by directly expressing the control law in the joint space, since manipulator joint limits are defined in this space. The subtask functions e_i used are computed from visual features:

$$e_i = s_i - s_i^* \tag{14.37}$$

where s_i is the current value of the visual features for subtask e_i and s_i^* their desired value. The interaction matrix L_{s_i} related to s_i is defined so that $\dot{s}_i = L_{s_i}\xi$, where ξ is the instantaneous camera velocity [8]. The interaction matrix L_{s_i} and the task Jacobian J_i are linked by the relation:

$$J_i = L_{s_i} M J_q, \tag{14.38}$$

where the matrix J_q denotes the robot Jacobian ($\dot{r} = J_q\dot{q}$) and M is the matrix that relates the variation of the camera velocity ξ to the variation of the chosen camera pose parametrization r ($\xi = M\dot{r}$) [8].

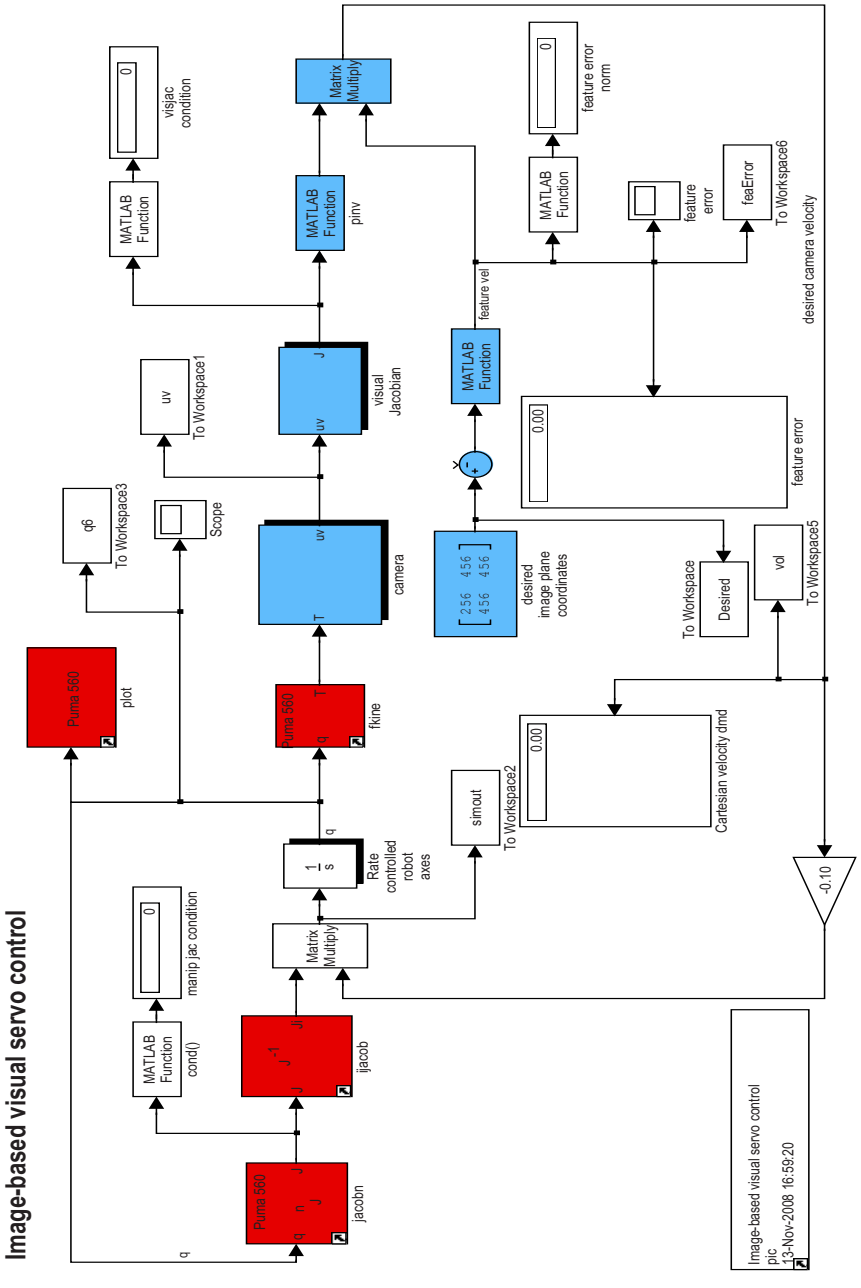
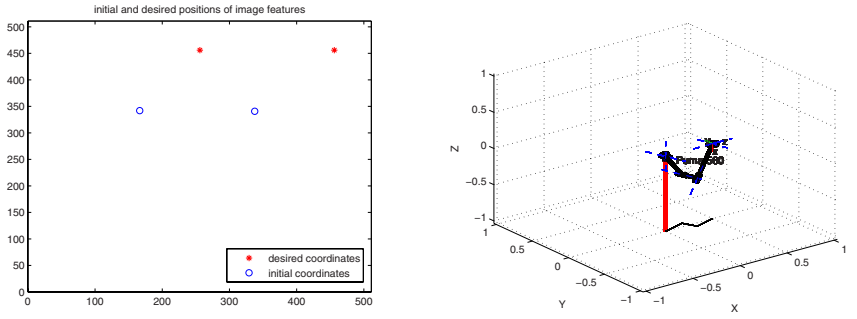
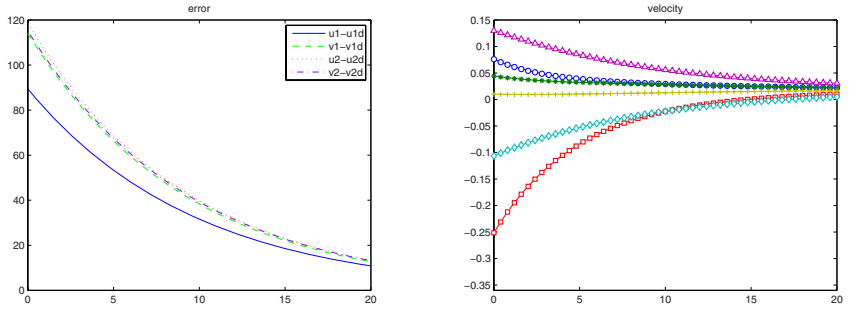


Fig. 14.1 Simulink® block diagram of IBVS control with joint limits



(a) (b)

Fig. 14.2: (a) Initial and final image feature locations, (b) the initial robot position



(a) (b)

Fig. 14.3: (a) Feature points motion, (b) and velocity of the camera

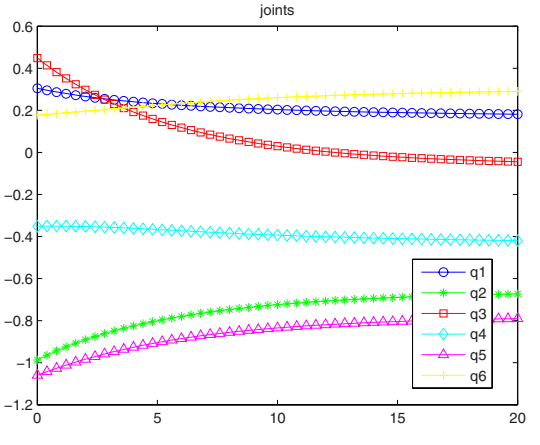


Fig. 14.4 Joint trajectories without joint limit avoidance

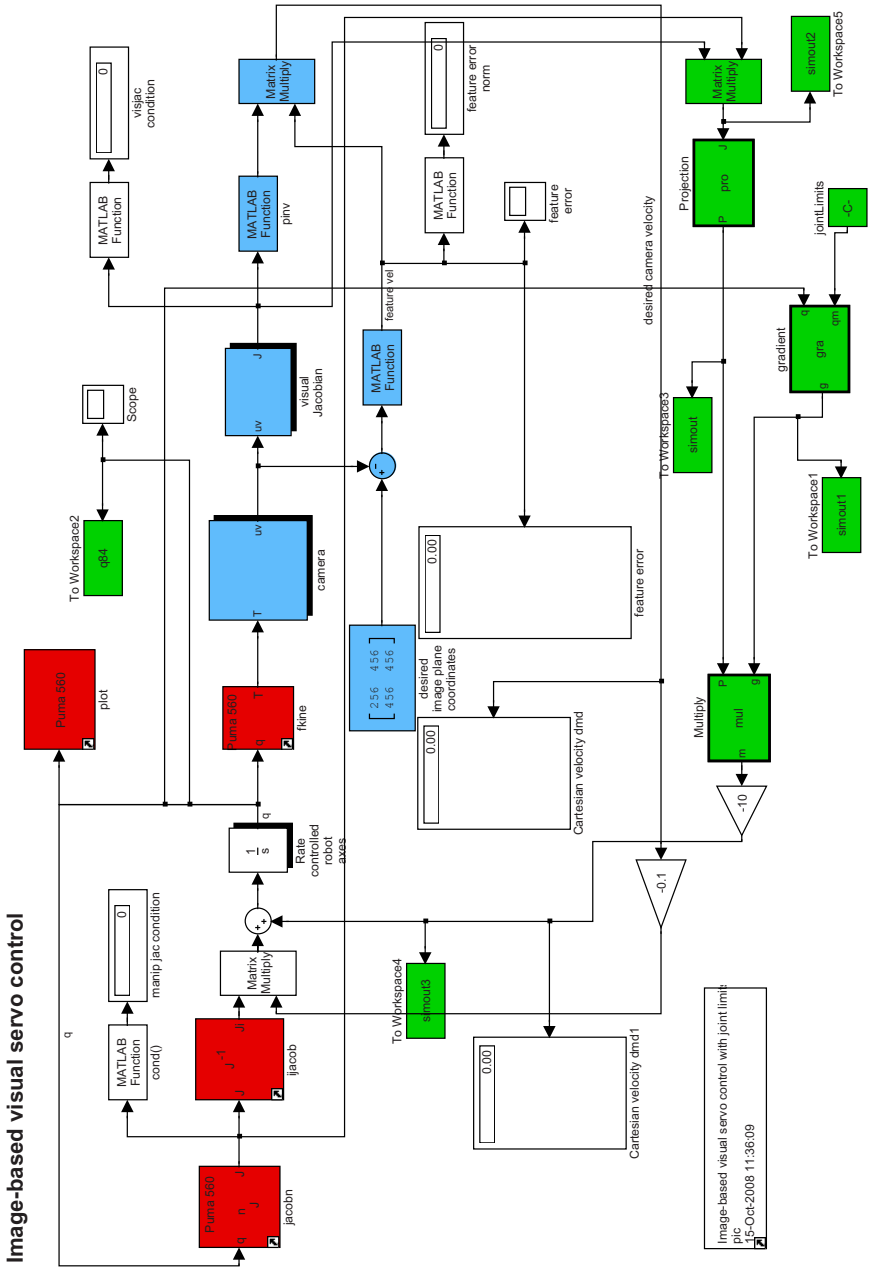


Fig. 14.5 Simulink® block diagram of IBVS control with joint limits

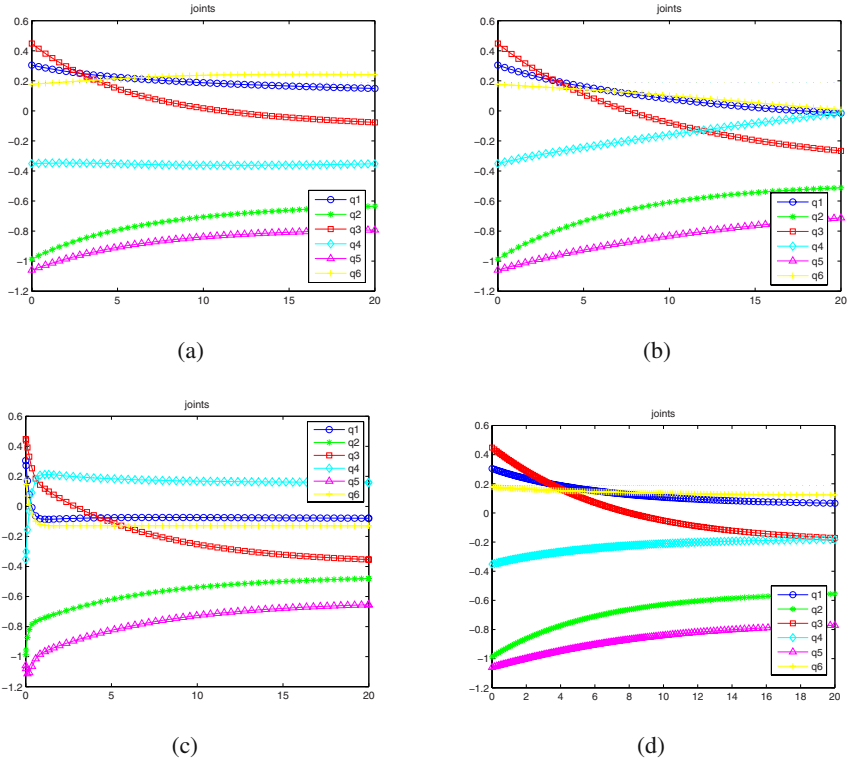


Fig. 14.6 Joints positions with control law for κ=0.1, 1, 100, auto

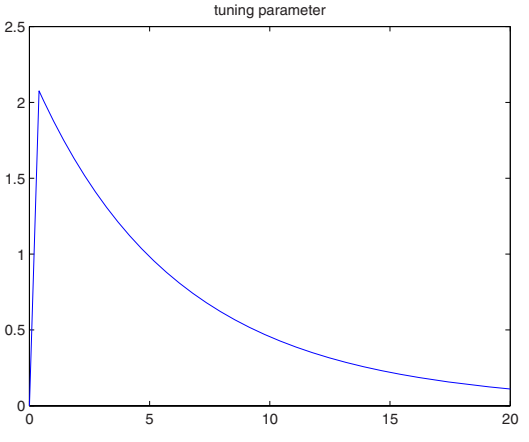


Fig. 14.7 The trajectory of tuning parameter using the automatic approach

The parameter that sets the amplitude of the control law due to the secondary task is very important. In (14.31), the secondary task may be insufficient to avoid a joint limit if κ is too small. Furthermore, if it is too large, κ will result in some overshoot in the effector velocity. One solution, proposed in [4], is to select the most critical axis and to compute automatically the minimum value of κ to stop any motion on this axis. More precisely, a critical axis is defined as an axis i such that q_i is between its joint limits and its activation threshold, and such that q_i approaches its joint limit due of the effect of the primary task. The effect of the primary task can be computed by performing a prediction step. Assuming that the robot is located in $q(t)$, without considering a secondary task, the predicted position $\hat{q}(t+1)$ is given by

$$\hat{q}(t+1) = q(t) + \dot{q}\Delta t. \tag{14.39}$$

From the set of all axes in the critical area such that q_i moves toward its joint limit (*i.e.*, $\hat{q}_i(t+1)$ is nearer to the corresponding joint limit than $q_i(t)$), the axis for which $\hat{q}_k(t+1)$ is the closest to its joint limits is selected. Then κ is computed to stop any motion on this axis (*i.e.*, $\hat{q}_k(t+1) - q_k(t) = 0$). The constraint $\Delta q_k = 0$ leads to compute κ as

$$\kappa = \frac{(\tilde{J}_1^+ \dot{e}_1)_k}{(P_1 g(q))_k}, \tag{14.40}$$

where k is the number of the closest axis from the joint limit and $(.)_k$ is the kth value of the vector.

14.3.2 Simulation Results

The whole of our simulations were conducted using Matlab® and the Machine Vision Toolbox [7] and Robotics Toolbox [6]. For each simulation, feature points consist of two points in 3D. This leaves the system with multiple DOF beyond that required to achieve the primary task (*i.e.*, the robot is redundant with respect to the given task).

Images were projected using a simulated camera with a focal length of 8×10^{-3} m. The u-coordinate (pix) and v-coordinate (pix) of the principal point are 256 and 256. The horizontal pixel pitch and vertical pixel pitch (pix/m) are 80000 and 80000.

The simulations use a 6 DOF eye-in-hand PUMA 560 robot. The standard PUMA robot device module (PUM) controls a 5 or 6 DOF PUMA-type robot. A PUMA robot usually consists of six revolute axes. The first three axes roughly correspond to a torso, shoulder, and elbow rotation. The last three axes implement a standard roll, pitch, roll wrist.

We assume that the camera velocity is the same as the end effector velocity ($M = I$). The estimate of depth of IBVS uses the depth value for the goal configuration. Considering only the main task, the block diagram used by Simulink® is shown in Figure 14.1.

Examples of the initial robot position and the initial and goal positions of feature points can be seen in Figure 14.2.

Figure 14.3 shows the result of a successful experiment if there is no joint limit for the robot. Figure 14.3(a–b) show the exponential decay of the error and the velocity of the robot, respectively. Though minor variations may arise, this 2D behavior is similar for all the successful experiments reported in this section. Figure 14.4 depicts the joint position of six axes of the robot.

Then we set the joint limits of q_6 to be smaller than 0.19. Therefore, if no particular strategy is considered to avoid joint limits, the visual task fails. Figure 14.5 shows the block diagram considering the joint limit. We performed a set of experiments using the cost function defined previously with various values of the coefficient κ (κ=0.1, κ=1, κ=100). During the experiments for these different values, if κ is too small (κ=0.1), the motion generated by the main task in the direction of the joint limits is not compensated enough by the secondary task, and then the joint limits cannot be avoided effectively. Figure 14.6(a) shows that q_6 exceeds the upper limit of the joint. However, if κ is too high (κ=100), it may result in too large velocities variance in the first seconds as shown in Figure 14.6(c). Tuning is therefore performed based on trial and error. This solution is not good.

The last plot (auto) on Figure 14.6(d) depicts the results obtained using the approach proposed where κ is automatically computed by (14.40). The change of κ with respect to the time is shown in Figure 14.7. It illustrates that the joint approach the joint limit and then move away from it.

14.4 Gradient Projection Method for Occlusion Avoidance

In this section, we describe how gradient projection methods can be used to prevent the occlusion of target features by static objects during the camera motion [11]. Below, we adopt a cost function proposed in [10], and develop the necessary methods for exploiting this cost function in a gradient projection framework.

14.4.1 Basic Algorithm Design

We assume here that image processing methods are available to detect the presence of a potential occluding object in the image. Given that the potential occlusion can be detected, let $d = \sqrt{d_x^2 + d_y^2}$ be the distance in the image from a feature point to the contour of the occluding object, and let x_a be the point of the occluding object that is the closest to the target in the image. The cost function V^{occ} is defined in the image space, so that it is maximal when d is 0, and nearly 0 when d is high. This is achieved by the cost function

$$V^{occ} = e^{-\beta d^2}. \tag{14.41}$$

The parameter β is arbitrary and can be used to tune the effect of the avoidance control law. The gradient in the image space is obtained by a simple calculation:

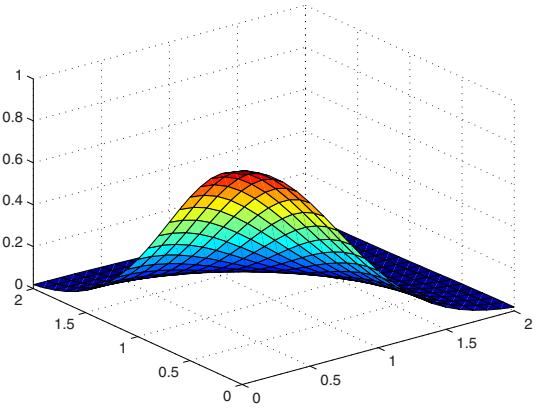


Fig. 14.8 Cost function used for visual-occlusion avoidance in the image space

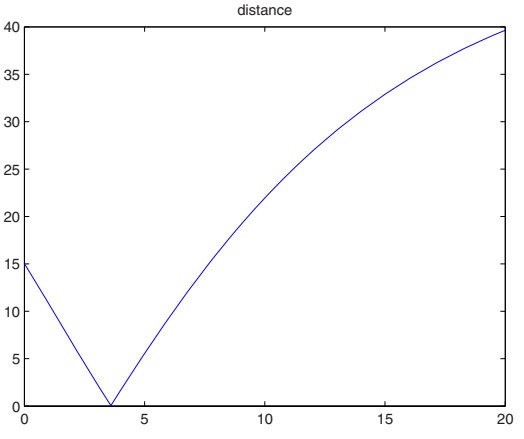


Fig. 14.9 Distance between the occluding point and target without occlusion avoidance

$$\nabla V^{occ} = \begin{pmatrix} -2\beta d_x e^{-\beta d^2} \\ -2\beta d_y e^{-\beta d^2} \end{pmatrix}. \tag{14.42}$$

The artificial force that avoids the occlusions can be now computed using (14.28). The transformation from the image space to the articular space is given by:

$$g^{occ} = -\left(\frac{\partial \Phi}{\partial r}\frac{\partial r}{\partial q}\right)^{+} \nabla_{\Phi} V^{occ} = -(L_{\Phi} M J_q)^{+} \nabla_{\Phi} V^{occ}, \tag{14.43}$$

where r is the chosen camera pose parametrization, M and J_q are the transformation matrices defined before, and L_{Φ} is the well-known interaction matrix related to the image point x_a.

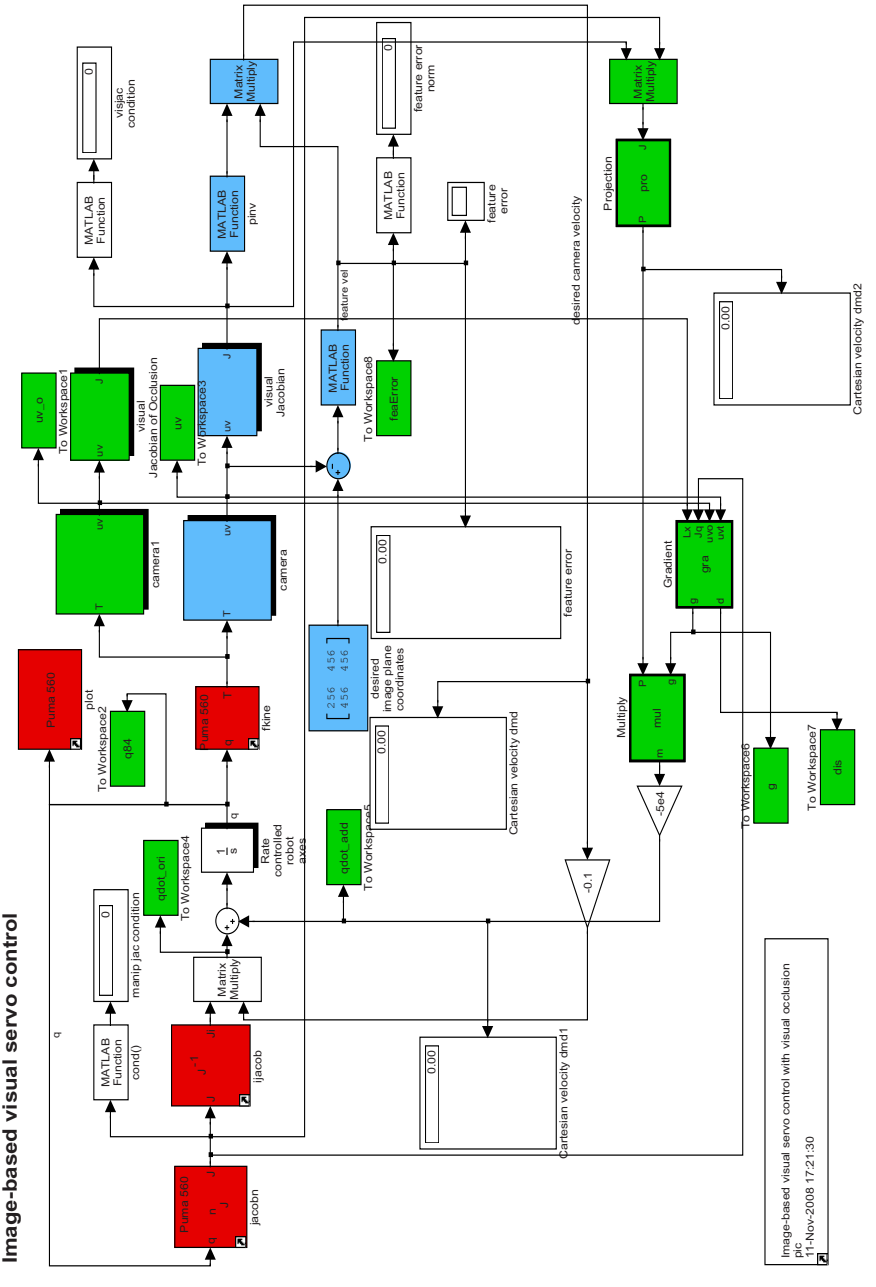


Fig. 14.10 Block diagram of IBVS control with occlusion avoidance

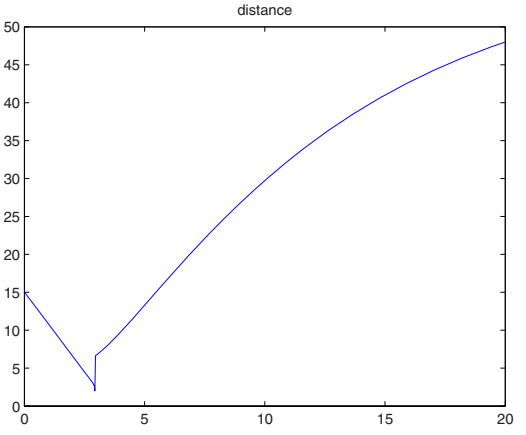


Fig. 14.11 Distance between the occluding point and target with occlusion avoidance

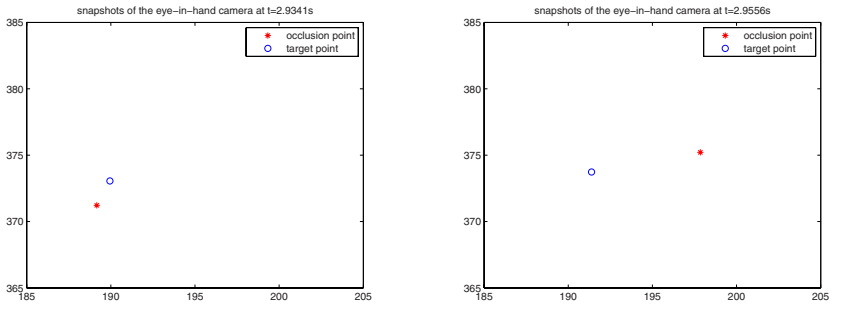
14.4.2 Simulation Results

These simulations use the same setup as described in Section 14.3.2. We place an object in the work space, and consider the possibility of occlusion of the first image feature by this object. The robot task is to reach a desired position, avoiding the visual occlusion. The model without considering the visual occlusion is the same as that in Section 14.3.2. The occluding point is set to be (1.0798 0.0649 0.0852), expressed in the camera frame. When the robot moves to the target position, the depth of the occluding point is 0.4248 according to the transform of the robot. The distance between the occluding point and the target is shown in Figure 14.9 without taking into account the environment constraints.

As shown on Figure 14.9, the distance decreases until reaching its minimum 0.0635 at $t = 3.6s$, and then increases afterward. It illustrates the target comes close to the occluding object and then is occluded by the object at $t = 3.6s$. After the visual occlusion, it moves away from the occluding object. Therefore, at $t = 3.6s$ the robot cannot observe the target since the occluding of a static point.

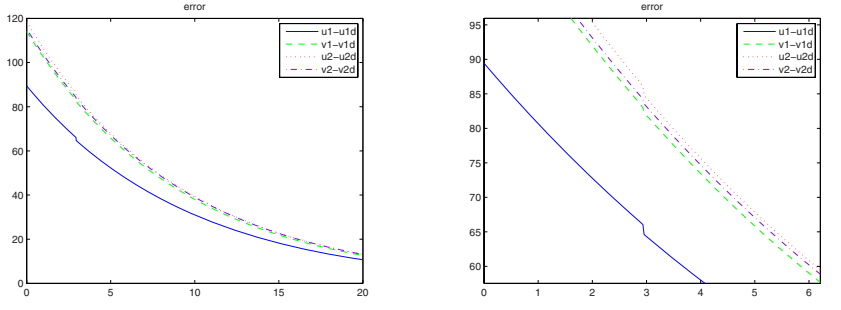
The block diagram with visual occlusion avoidance is given in Figure 14.10. When considering the visual occlusion, the corresponding distance between occluding point and the target after adding occlusion avoidance are given in Figure 14.11. At the beginning, the target moves close to the occluding point as shown previously. But after it reaches 1.9999 at $t = 2.9341s$, the distance jumps to 6 in the next iteration. The force generated by the gradient of the cost function drives the robot away from the occluding object after it is predicted.

Figures 14.12 and 14.13 show respectively some screen shots taken from the simulation during the servo, the points trajectory in the image plane, and the error of the main task. Figure 14.12(a) shows the image when occlusion is predicted at $t = 2.9341s$, and Figure 14.12(b) is the one when occlusion is avoided. Figure 14.13(a) shows the decay of the error of the main task. Through zooming in Figure 14.13(a),



(a) (b)

Fig. 14.12 Snapshots of the eye-in-hand camera



(a) (b)

Fig. 14.13 Feature points motions

the detailed trajectory of the features error around $t = 2.9341s$ is shown in Figure 14.13(b). We can see that the error still decreases exponentially, except the sawtooth around $t = 2.9341s$, which corresponds to the activation of the artificial force.

14.5 Gradient Projection Method for Keeping Target Features in the Field of View

In this section, we describe how gradient projection methods can be used to keep target features in the camera field of view throughout the camera motion. Below, we adopt a cost function proposed in [10], and describe the necessary methods for exploiting this cost function in a gradient projection framework.

14.5.1 Basic Algorithm Design

One approach to keeping target features in the field of view is to construct a cost function that takes a value equal to zero when target features are at the image center,

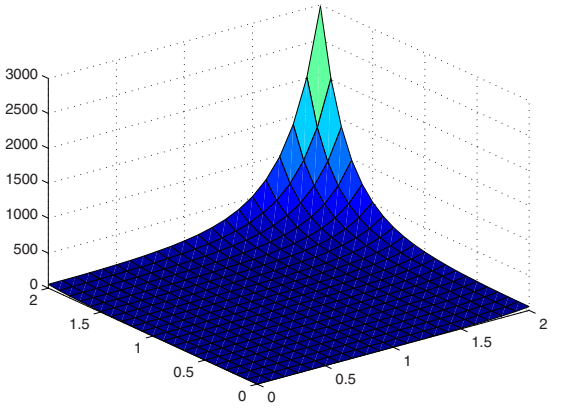


Fig. 14.14 Cost function of the field of view constraint in the image space

and which grows exponentially as target features move toward the image boundary. Such a function has been proposed in [10]

$$V^{fv} = \alpha e^{\beta d^2}, \tag{14.44}$$

where d is the distance between the middle of the image and the object of interest, and α and β are parameters to tune the effect of the avoidance.

The gradient in the image space is given by

$$\nabla V^{fv} = \begin{pmatrix} 2\beta d_x e^{\beta d^2} \\ 2\beta d_y e^{\beta d^2} \end{pmatrix}. \tag{14.45}$$

The artificial force to avoid the object beyond the boundary of the image plane is computed by (14.28). It is transformed from the image space to the articular space using

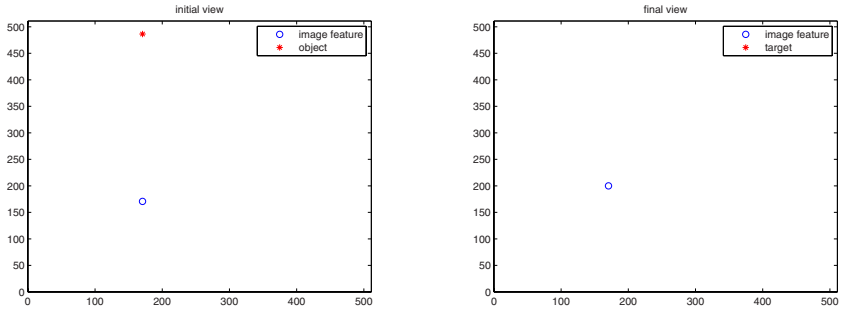
$$g^{fv} = -\left(\frac{\partial \Phi}{\partial r}\frac{\partial r}{\partial q}\right)^{+} \nabla_{\Phi} V^{fv} = -(L_{\Phi} M J_q)^{+} \nabla_{\Phi} V^{fv}, \tag{14.46}$$

where M and J_q are the transformation matrices defined as in (14.43) and L_{Φ} is the interaction matrix related to the center of the image plane.

14.5.2 Simulation Results

In this set of simulations, the task is to control the position of a feature point in the image while keeping a second feature point (the object of interest) within the field of the view of the camera.

Figure 14.15(a) depicts the initial image from the camera. The object of interest is very close to the border of the image. In this case, the object will move out of the



(a) (b)

Fig. 14.15 Initial and final view of the image feature and the object without considering field of view constraints

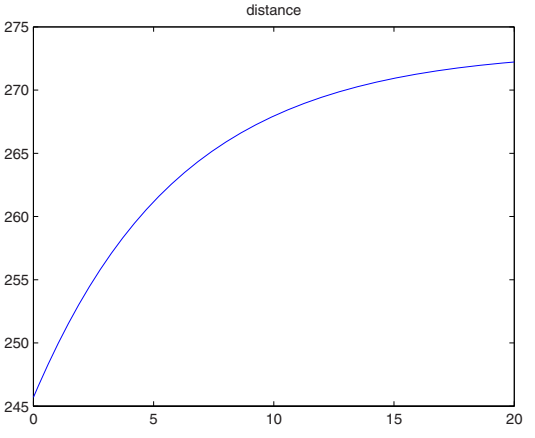


Fig. 14.16 Distance between the object and the center of the image without using field of view constraints

field of view of the camera at the end of the task. As shown on Figure 14.15(b), we cannot see the object in the camera view during the servoing without using the field of view constraint. The distance between the object and the center of the image is shown in Figure 14.16 without taking into account the constraints.

To realize the objective of keeping the object of interest in the field of view during the control, redundancy of the robot is applied to the servoing. The initial view of the camera is the same as above, and the final view is shown on Figure 14.17. The object of interest remains in the view of camera. The corresponding distance between the object and center of the image after using field of view constraints are given in Figure 14.18. When the object moves closed to the border of the camera view, the

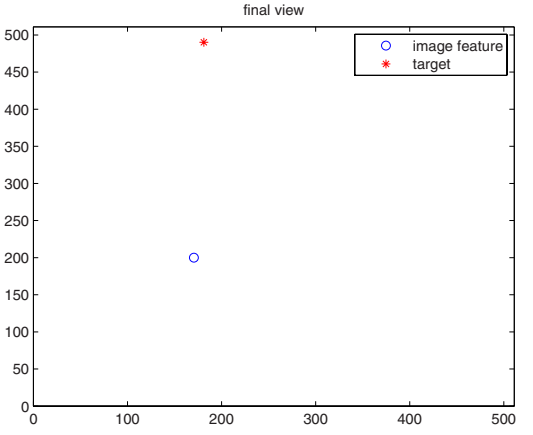


Fig. 14.17 Final view of the image feature and the object

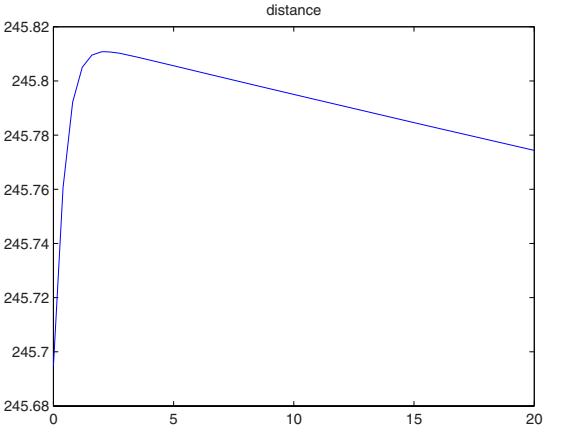


Fig. 14.18 Distance between the object and center of the image with field of view constraints

artificial force starts to decrease the distance and constrains the object remain in the image of the camera.

14.6 Conclusions

This chapter describes the performance of gradient projection methods to satisfy three kinds of constraints that typically arise in visual servo applications: joint limits, visual occlusion, keeping the desired objective inside view. The task function approach provides a formalism by which a secondary task can be achieved without disturbing the execution of the primary task. With gradient projection methods, the secondary task corresponds to the gradient of a cost function, and this gradient is

projected onto the null space of the primary task to yield motions that will honor the constraints without interfering with the performance of the primary task. The approach was tested in simulation for each of the three types of constraints, and found to perform reasonably well, even though the method is sensitive to the selection of the parameter that influences the relative weight for velocity components assigned to the primary and secondary tasks.

References

[1] Bertsekas, D.P.: Nonlinear Programming. Athena Scientific, Belmont (1995)
[2] Chaumette, F., Hutchinson, S.: Visual servo control, part I: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)
[3] Chaumette, F., Hutchinson, S.: Visual servo control, part II: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)
[4] Chaumette, F., Marchand, E.: A redundancy-based iterative approach for avoiding joint limits: application to visual servoing. IEEE Trans. on Robotics and Automation 17(5), 719–730 (2001)
[5] Chiacchio, P., Chiaverini, S.: Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy. Int. J. Robot. Res. 10, 410–425 (1991)
[6] Corke, P.: A robotics toolbox for MATLAB. IEEE Robotics and Automation Magazine 3(1), 24–32 (1996)
[7] Corke, P.: Machine vision toolbox. IEEE Robotics and Automation Magazine 12(4), 16–25 (2005)
[8] Espiau, B., Chaumette, F., Rives, P.: A New Approach to Visual Servoing in Robotics. IEEE Trans. on Robotics and Automation 8, 313–326 (1992)
[9] Mansard, N., Chaumette, F.: Task sequencing for high-level sensor-based control. IEEE Trans. on Robotics 23(1), 60–72 (2007)
[10] Marchand, E., Hager, G.: Dynamic sensor planning in visual servoing. In: IEEE Int. Conf. on Robotics and Automation (ICRA 1998), Leuven, Belgium, pp. 1988–1993 (1998)
[11] Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Trans. on Robotics and Automation 18(4), 534–549 (2002)
[12] Nakamura, Y.: Advanced robotics: redundancy and optimization. Addison-Wesley publishing company, Reading (1991)
[13] Rosen, J.B.: The gradient projection method for nonlinear programming. part 1: Linear constraints. J. SIAM 8, 181–217 (1960)
[14] Samson, C., Le Borgne, M., Espiau, B.: Robot Control: The Task Function Approach. Clarendon Press, Oxford (1992)
[15] Siciliano, B.: Kinematic control of redundant manipulators: A tutorial. Journal of Intelligent and Robotic Systems 3, 201–212 (1990)
[16] Yoshikawa, T.: Foundations of robotics: analysis and control. MIT Press, Cambridge (1990)

Chapter 15
Image-based Visual Servo Control Design with Multi-Constraint Satisfaction

Sophie Tarbouriech and Philippe Souères

Abstract. This chapter presents an application of advanced control techniques to the design of a multicriteria image-based controller for robotics application. The problem of positioning a 3 degrees of freedom (DOF) camera with respect to a mobile visual target with unknown square integrable velocity is considered. The proposed controller allows to stabilize the camera and determine the associated region of stability in spite of unknown values of the target points depth, bounds on admissible visual features errors to guarantee visibility, and limits on the camera velocity and acceleration. The description of the closed-loop system is based on a mixed polytopic and norm-bounded representation of uncertainties combined with an original sector condition. With this representation, the unknown target velocity is considered as a disturbance limited in energy and linear matrix inequality (LMI)-based optimization schemes are used to compute the feedback gain that leads to the maximization of the size of the region of stability associated to the closed-loop system. Two applications of the method are studied and simulated by considering a model of car-like robot equipped with proximity sensors and supporting a camera mounted on a plan-platform. The interest of the control approach for linking dynamically a sequence of sensor-based navigation tasks for a mobile robot is illustrated.

15.1 Introduction

Image-based visual servoing aims at controlling robotic systems by regulating feature errors directly in the camera image-plane. In order to deal with parameter uncertainty, and take into account additional constraints such as visibility or actuators limitation, the design of robust control schemes turns out to be essential [5]. As shown in [14], uncertainty in the depth distribution of target points may strongly

Sophie Tarbouriech and Philippe Souères
LAAS-CNRS, Université de Toulouse, 7 Avenue du Colonel Roche,
31077 Toulouse cedex 4, France
e-mail: {tarbour,soueres}@laas.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 275–294.
springerlink.com

reduce the domain of stability of the system. However, in most parts of applications of image-based control the depth of the target points is unknown, and the interaction matrix is computed at the final position [4]. An adaptive 2D-vision based servo controller robust with respect to depth measurement was proposed in [24]. Ensuring the visibility during the motion is also an important issue. Different tracking methods that combine geometric information with local image measurements have been proposed to this end in [15]. The idea of mixing feature-based control with path-planning in the image was proposed in [16]. Robust control is particularly interesting when dealing with moving targets for which the unknown velocity can be bounded in some sense. In [9], generalized predictive control (GPC) and $\mathscr{H}_\infty$ fast visual servoing were proposed for tracking moving target points with a medical manipulator. Advanced control techniques allow to consider different kinds of constraints simultaneously at the control synthesis level. A general framework for image-based positioning control design, based on robust quadratic methods and differential inclusions, was proposed in [21]. The main drawback of this approach was that the conditions were given in the form of bilinear matrix inequalities. A unified approach for position-based and image-based control, through rational systems representation, was also proposed in [1]. Other recent methods were published, as for instance switching strategies for ensuring visibility constraint [6], generation of circular trajectories for minimizing the trajectory length [7], use of catadioptric camera [13].

In this chapter, techniques allowing to design a multicriteria image-based controller in order to track moving targets with square integrable velocity are proposed. The considered approach is based on both polytopic and norm-bounded representations of uncertainties and uses an original sector condition for the description of saturation terms. The proposed controller allows to stabilize the camera despite unknown values of the target points depth, bounds on admissible visual features errors to ensure visibility, and limits on the camera velocity and acceleration. Lyapunov analysis and LMI-based optimization procedures are used to characterize a compact neighborhood of the origin inside which the system trajectories remain during the tracking. When the target is at rest, the asymptotic stability of the closed-loop system is proved and a maximal stability domain is determined.

The chapter is organized as follows. Section 15.2 precisely describes the system under consideration and formally states the control design problem. In Section 15.3, some preliminary results exhibiting useful properties of the closed-loop system are presented. Section 15.4 is devoted to control design results. Section 15.5 addresses simulation results regarding the case of a camera mounted on a pan-platform supported by a wheeled robot. Finally, some concluding remarks together with forthcoming issues end the chapter.

Notation. For $x, y \in \Re^n$, $x \succeq y$ means that $x_{(i)} - y_{(i)} \geq 0, \forall i = 1, \ldots, n$. The elements of $A \in \Re^{m \times n}$ are denoted by $A_{(i,l)}$, $i = 1, \ldots, m$, $l = 1, \ldots, n$. $A_{(i)}$ denotes the ith row of A. For two symmetric matrices, A and B, $A > B$ means that $A - B$ is positive definite. A' denotes the transpose of A. $D(x)$ denotes a diagonal matrix obtained from vector x. 1_m denotes the m-order vector $1_m = [1 \ldots 1]' \in \Re^m$. I_m denotes the

m-order identity matrix. $Co\{.\}$ denotes a convex hull. For $u \in \Re^m$, $sat_{u_0}(u_{(i)}) = sign(u_{(i)}) \min(|u_{(i)}|, u_{0(i)})$ with $u_{0(i)} > 0$, $i = 1, ..., m$.

15.2 Problem Statement

The problem of positioning a camera with respect to a mobile visual target is addressed. The system under consideration consists of a camera which is supported by a robotic system. The camera is free to execute any horizontal translations and rotations about the vertical axis.

The objective of the chapter is to design a controller to stabilize the camera with respect to the target despite the following conditions:

- *(C1)* the depth of the target points with respect to the camera frame, is bounded but unknown;
- *(C2)* the visual signal errors, in the image, must remain bounded during the stabilization process to ensure visibility;
- *(C3)* the velocity and the acceleration of the camera remain bounded to satisfy the limits on the actuators dynamics; and
- *(C4)* the velocity vector of the target is supposed to be square integrable but unknown.

15.2.1 System Description

Let us now describe more precisely the system under consideration. The target is made of three points E_i, $i = 1, 2, 3$, equispaced on a horizontal line, and located at the same height as the camera optical center C. Let R be a frame attached to the scene and R_c a frame attached to the camera, having its origin at C and its z-axis directed along the optical axis. Let $T \in \Re^3$ denote the reduced kinematic screw of the camera which expresses the translational and rotational velocities of R_C with respect to R. The target is assumed to move as if it was fixed to a unicycle: it can rotate about its central point E_2 whereas its linear velocity v_E is always perpendicular to the line (E_1, E_3) as indicated in Figure 15.1. ω_E denotes the target rotational velocity.

Furthermore, we make the hypothesis that the camera intrinsic parameters are known and consider the metric pinhole model with focal length $f = 1$.

We denote by $l > 0$ the distance between the target points, α the angle between the target line and the optical axis, and η the angle between the optical axis and the line (CE_2) (see Figure 15.1). Without lost of generality we will assume that the camera is initially located in the left-half-plane delimited by the line (E_1, E_3) and that the distance $d = CE_2$ is bounded as

$$d \in [d_{min}, d_{max}]. \tag{15.1}$$

Furthermore, to prevent from projection singularities, it is considered that

$$\alpha \in [-\pi + \alpha_{min}, -\alpha_{min}] \tag{15.2}$$

where $\alpha_{min} > 0$ is a small angle.

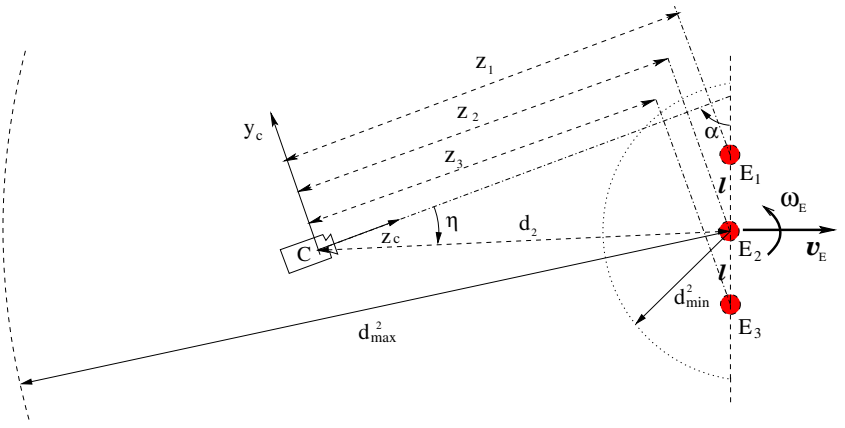


Fig. 15.1 Description of the vision-based task

For $i = 1,2,3$, let us denote respectively by Y_i and Y_i^* the ordinates of the current and desired target points in the image. Following the formalism introduced in [10], a simple choice for the positioning task function is

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \in \Re^3$$

with

$$e_i = Y_i - Y_i^*.$$

The desired camera position corresponds to $Y_2^* = 0$ and $Y_3^* = -Y_1^*$. In addition, to guarantee the visibility of the target, the condition C2 is specified by imposing the following bound on task components:

$$|e_i| \leq \beta,\ i = 1,2,3\ ;\ \text{with}\ \beta > 0. \tag{15.3}$$

Consequently, the angle η is bounded by

$$|\eta| \leq \eta_{max} = \arctan(\beta) < \pi/2, \tag{15.4}$$

and therefore the depth $z_2 = d\cos(\eta)$ of the central point E_2 is bounded as

$$z_2 \in [d_{min}\cos(\eta_{max}), d_{max}]. \tag{15.5}$$

The relation between the time-derivative of the task function and the kinematic screw is given by the optical flow equations:

$$\dot{e} = L(z,e)T + \frac{\partial e}{\partial t} \tag{15.6}$$

where $L(z,e) \in \Re^{3\times 3}$ is the image Jacobian

$$L(z,e) = \begin{bmatrix} -1/z_1 & (e_1+Y_1^\star)/z_1 & 1+(e_1+Y_1^\star)^2 \\ -1/z_2 & (e_2+Y_2^\star)/z_2 & 1+(e_2+Y_2^\star)^2 \\ -1/z_3 & (e_3+Y_3^\star)/z_3 & 1+(e_3+Y_3^\star)^2 \end{bmatrix}. \tag{15.7}$$

The term $\frac{\partial e}{\partial t}$, which represents task variation due to the target motion, can be expressed by the relation

$$\frac{\partial e}{\partial t} = B(z,e)\omega, \tag{15.8}$$

where the matrix $B(z,e) \in \Re^{3\times 4}$ is defined by

$$B(z,e) = \begin{bmatrix} 1/z_1 & -(e_1+Y_1^\star)/z_1 & 1/z_1 & -(e_1+Y_1^\star)/z_1 \\ 1/z_2 & -(e_2+Y_2^\star)/z_2 & 0 & 0 \\ 1/z_3 & -(e_3+Y_3^\star)/z_3 & -1/z_3 & (e_3+Y_3^\star)/z_3 \end{bmatrix} \tag{15.9}$$

and

$$\omega = \begin{bmatrix} -v_E\cos(\alpha) \\ -v_E\sin(\alpha) \\ -l\omega_E\cos(\alpha) \\ -l\omega_E\sin(\alpha) \end{bmatrix} \in \Re^4. \tag{15.10}$$

The target velocity vector $\begin{bmatrix} v_E & \omega_E \end{bmatrix}'$ is supposed to be square integrable but unknown. The vector ω can then be considered as a disturbance vector and $B(z,e)$ as the associated disturbance matrix. With this notation, the condition C4 can be specified as follows: $\omega \in \mathscr{L}_2$ and there exists a finite scalar $\delta_1 > 0$ such that

$$\|\omega\|_2^2 = \int_0^\infty \omega(\tau)'\omega(\tau)d\tau = \int_0^\infty (v_E^2(\tau) + l^2\omega_E^2(\tau))d\tau \le \frac{1}{\delta_1}. \tag{15.11}$$

To take into account the limits on the actuators dynamics, the statement of condition C3 can be specified by introducing the following bounds on the camera velocity and acceleration:

$$-u_1 \preceq T \preceq u_1, \tag{15.12}$$

$$-u_0 \preceq \dot{T} \preceq u_0. \tag{15.13}$$

By considering the extended state vector

$$x = \begin{bmatrix} e \\ T \end{bmatrix} \in \Re^6, \tag{15.14}$$

with the following matrices

$$\begin{aligned} \mathbb{A}(z,x) &= \begin{bmatrix} 0 & L(z,e) \\ 0 & 0 \end{bmatrix} \in \Re^{6\times 6}; \\ \mathbb{B}_1 = \begin{bmatrix} 0 \\ I_3 \end{bmatrix} &\in \Re^{6\times 3};\ \mathbb{B}_2(z,x) = \begin{bmatrix} B(z,e) \\ 0 \end{bmatrix} \in \Re^{6\times 4} \end{aligned} \tag{15.15}$$

the system under consideration reads

$$\dot{x} = \mathbb{A}(z,x)x + \mathbb{B}_1\dot{T} + \mathbb{B}_2(z,x)\omega, \tag{15.16}$$

where the acceleration of the camera $\dot{T}$ is the control vector.

15.2.2 Problem Formulation

To cope with constraint (15.13), the control law we consider has the form

$$\dot{T} = sat_{u_0}(\mathbb{K}x) \text{ with } \mathbb{K} = \begin{bmatrix} K_1 & K_2 \end{bmatrix} \in \Re^{3\times 6}. \tag{15.17}$$

Hence, the closed-loop system reads

$$\dot{x} = \mathbb{A}(z,x)x + \mathbb{B}_1 sat_{u_0}(\mathbb{K}x) + \mathbb{B}_2(z,x)\omega. \tag{15.18}$$

Relative to the closed-loop system (15.18), one has to take into account the constraints (15.3) and (15.12), which means that the state x must belong to the polyhedral set

$$\Omega(x) = \left\{ x \in \Re^6; -\begin{bmatrix} \beta 1_3 \\ u_1 \end{bmatrix} \preceq x \preceq \begin{bmatrix} \beta 1_3 \\ u_1 \end{bmatrix} \right\}. \tag{15.19}$$

The problem we intend to solve with respect to the closed-loop system (15.18), subject to constraints (15.19), can be summarized as follows.

Problem 15.1. Determine a gain $\mathbb{K}$ and two regions $\mathscr{S}_0$ and $\mathscr{S}_1$, as large as possible, such that, in spite of conditions C1 and C4, the constraints C2 and C3 are satisfied and:

- *(internal stability)* when $\omega = 0$, for any $x(0) \in \mathscr{S}_0$ the closed-loop trajectories of system (15.18) converge asymptotically to the origin; and
- *(input-to-state stability)* when $\omega \neq 0$, the closed-loop trajectories remain bounded in $\mathscr{S}_1$ for any $x(0) \in \mathscr{S}_1$ and for all ω satisfying (15.11).

The first item of Problem 15.1 corresponds to address the case of a fix target [22], [11]. Indeed, this case consists of stabilizing a camera in front of a target, and determining the associated region of stability in spite of unknown value of the target points depth, bounds on admissible feature errors which guarantee visibility, and limits on the camera velocity and acceleration. The second item of Problem 15.1 corresponds to the tolerance of the closed-loop system with respect to the perturbation ω. In this case, the objective is to ensure that the closed-loop trajectories are bounded for any disturbance ω satisfying (15.11).

15.3 Preliminary Results

Let us specify some properties of the closed-loop system (15.18). By defining the decentralized dead-zone nonlinearity $\phi(\mathbb{K}x) = sat_{u_0}(\mathbb{K}x) - \mathbb{K}x$, defined by

$$\phi(\mathbb{K}_{(i)}x) = \begin{cases} u_{0(i)} - \mathbb{K}_{(i)}x & \text{if } \mathbb{K}_{(i)}x > u_{0(i)} \\ 0 & \text{if } |\mathbb{K}_{(i)}x| \leq u_{0(i)} \\ -u_{0(i)} - \mathbb{K}_{(i)}x & \text{if } \mathbb{K}_{(i)}x < -u_{0(i)} \end{cases} \tag{15.20}$$

$\forall i = 1,2,3$, the closed-loop system (15.18) can be described by

$$\dot{x} = (\mathbb{A}(z,x) + \mathbb{B}_1\mathbb{K})x + \mathbb{B}_1\phi(\mathbb{K}x) + \mathbb{B}_2(z,x)\omega. \tag{15.21}$$

Consider a matrix $\mathbb{G} \in \Re^{3\times 6}$ and define the set

$$S(u_0) = \{x \in \Re^6; -u_0 \preceq (\mathbb{K} - \mathbb{G})x \preceq u_0\}. \tag{15.22}$$

The following lemma, issued from [12], can then be stated.

Lemma 15.1. *Consider the nonlinearity $\phi(\mathbb{K}x)$ defined in (15.20). If $x \in S(u_0)$, then the relation*

$$\phi(\mathbb{K}x)'M(\phi(\mathbb{K}x) + \mathbb{G}x) \leq 0 \tag{15.23}$$

is satisfied for any diagonal positive definite matrix $M \in \Re^{3\times 3}$.

Let us first write in a more tractable way matrices $\mathbb{A}(z,x)$ and $\mathbb{B}_2(z,x)$, and therefore the closed-loop system (15.21). In this sense, we define the matrices

$$B_1(z) = \begin{bmatrix} -1/z_1 & Y_1^*/z_1 & 1 + (Y_1^*)^2 \\ -1/z_2 & Y_2^*/z_2 & 1 + (Y_2^*)^2 \\ -1/z_3 & Y_3^*/z_3 & 1 + (Y_3^*)^2 \end{bmatrix};$$

$$B_2(z) = \begin{bmatrix} 1/z_1 & 0 & 0 \\ 0 & 1/z_2 & 0 \\ 0 & 0 & 1/z_3 \end{bmatrix}; B_3 = \begin{bmatrix} 2Y_1^* & 0 & 0 \\ 0 & 2Y_2^* & 0 \\ 0 & 0 & 2Y_3^* \end{bmatrix};$$

$$B_4(z) = \begin{bmatrix} 1/z_1 & -Y_1^*/z_1 & 1/z_1 & -Y_1^*/z_1 \\ 1/z_2 & -Y_2^*/z_2 & 0 & 0 \\ 1/z_3 & -Y_3^*/z_3 & -1/z_3 & Y_3^*/z_3 \end{bmatrix};$$

$$B_5(z) = \begin{bmatrix} 0 & -1/z_1 & 0 & -1/z_1 \\ 0 & -1/z_2 & 0 & 0 \\ 0 & -1/z_3 & 0 & 1/z_3 \end{bmatrix}; D(e) = \begin{bmatrix} e_1 & 0 & 0 \\ 0 & e_2 & 0 \\ 0 & 0 & e_3 \end{bmatrix};$$

$$\mathbb{R} = \begin{bmatrix} I_3 & 0 \end{bmatrix}; \mathbb{C} = \begin{bmatrix} 0 & I_3 \end{bmatrix}.$$

Then, the closed-loop system reads

$$\begin{aligned} \dot{x} = (\mathbb{R}'B_1(z)\mathbb{C} + \mathbb{R}'T_{(2)}B_2(z)\mathbb{R} + \mathbb{B}_1\mathbb{K} + \mathbb{R}'T_{(3)}(B_3 + D(e))\mathbb{R})x + \mathbb{B}_1\phi(\mathbb{K}x) \\ + \mathbb{R}'(B_4(z) + D(e)B_5(z))\omega. \end{aligned} \tag{15.24}$$

According to the notation previously specified, $T_{(2)}$ and $T_{(3)}$ respectively denote the 2th and 3th components of vector T. Furthermore, in (15.6) and (15.7), the constraints on the vector $z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}' \in \Re^3$ follow the description given in Figure 15.1. From relations (15.1), (15.2), (15.4) and (15.5), the depth of target points E_1 and E_3 can be expressed in terms of the depth z_2 of the central point E_2:

$$z = \begin{bmatrix} z_2 + l\cos(\alpha) \\ z_2 \\ z_2 - l\cos(\alpha) \end{bmatrix}.$$

It is important to remark that by definition one gets $l << z_2$. Hence, the following approximation can be done:

$$\begin{array}{l} \frac{1}{z_1} \simeq \frac{1}{z_2}(1 - \frac{l\cos(\alpha)}{z_2}) = \frac{1}{z_2} - \frac{l\cos(\alpha)}{z_2^2} = p_1 - p_2; \\ \frac{1}{z_2} = p_1; \\ \frac{1}{z_3} \simeq \frac{1}{z_2}(1 + \frac{l\cos(\alpha)}{z_2}) = \frac{1}{z_2} + \frac{l\cos(\alpha)}{z_2^2} = p_1 + p_2. \end{array} \tag{15.25}$$

From the definition of the admissible intervals relative to z_2 and α given by (15.2) and (15.5), it follows that the scalars p_1 and p_2 in (15.25) satisfy

$$p_{jmin} \leq p_j \leq p_{jmax},\ j = 1,2,$$

with the bounds

$$\begin{array}{ll} p_{1min} = \dfrac{1}{d_{2max}}; & p_{1max} = \dfrac{1}{d_{2min}cos(\eta_{max})}; \\ p_{2min} = \dfrac{lcos(-\pi + \alpha_{min})}{(d_{2min}cos(\eta_{max}))^2}; & p_{2max} = \dfrac{lcos(-\alpha_{min})}{(d_{2min}cos(\eta_{max}))^2}. \end{array} \tag{15.26}$$

Thus, the matrices $L(z,e)$ and $B(z,e)$, and therefore the matrices $\mathbb{A}(z,x)$ and $\mathbb{B}_2(z,x)$, depend on two uncertain parameters p_1 and p_2. Thus, from (15.25), $B_1(z)$, $B_2(z)$, $B_4(z)$ and $B_5(z)$ depend on two uncertain parameters p_1 and p_2. By using the classical framework of uncertain systems [2], it follows that these matrices belong to a polytope with 4 vertices given by the combinations of value of p_1 and p_2 in their definition interval:

$$B_k(z) \in Co\{B_{kj}, j = 1,...,4\},\ \text{for}\ k = 1,2,4,5. \tag{15.27}$$

Consequently, the closed-loop system (15.21) or (15.24) can be written as

$$\begin{array}{l} \dot{x} = \sum_{j=1}^{4} \lambda_j \{[\mathbb{A}_{1j} + \mathbb{R}'T_{(2)}B_{2j}\mathbb{R} + \mathbb{R}'T_{(3)}(B_3 + D(e))\mathbb{R} + \mathbb{B}_1\mathbb{K}]x \\ \qquad + \mathbb{R}'(B_{4j} + D(e)B_{5j})\omega\} + \mathbb{B}_1\phi(\mathbb{K}x), \end{array} \tag{15.28}$$

with $\sum_{j=1}^{4} \lambda_j = 1$, $\lambda_j \geq 0$ and $\mathbb{A}_{1j} = \mathbb{R}'B_{1j}\mathbb{C}$.

15.4 Control Design Results

This section is dedicated to the presentation of the main results of the chapter. Existence conditions are provided through Theorem 15.1 and Proposition 15.1. Then,

the choice of some particular solutions is obtained thanks to convex optimization schemes.

15.4.1 Theoretical Issues

Let us now consider a positive definite function $V(x) > 0, \forall x \neq 0$, with $V(0) = 0$, such that its time-derivative along the trajectories of the closed-loop saturating system (15.21) verifies

$$\dot{V}(x) \leq \omega' \omega, \tag{15.29}$$

for all ω satisfying (15.11) and $\forall x \in \mathscr{D}$, where $\mathscr{D}$ is some bounded domain inside the basin of attraction of (15.21). If (15.29) is verified, it follows that

$$V(x(t)) - V(x(0)) \leq \int_0^t \omega(\tau)' \omega(\tau) d\tau, \tag{15.30}$$

$\forall x \in \mathscr{D}$ and $\forall t \geq 0$. From (15.11), we have $V(x(t)) \leq V(x(0)) + \frac{1}{\delta_1}$ and, hence,

$$V(x(0)) \leq \frac{1}{\zeta} \Longrightarrow V(x(t)) \leq \frac{1}{\zeta} + \frac{1}{\delta_1}. \tag{15.31}$$

Define the sets $\mathscr{S}_1 = \{x \in \Re^6; V(x) \leq \frac{1}{\zeta} + \frac{1}{\delta_1}\}$ and $\mathscr{S}_0 = \{x \in \Re^6; V(x) \leq \frac{1}{\zeta}\}$. Provided that $\mathscr{S}_1$ is included in the basin of attraction of (15.21), from (15.31) it follows that the closed-loop system trajectories remain bounded in $\mathscr{S}_1$, for any initial condition $x(0) \in \mathscr{S}_0$, and any ω satisfying (15.11) [3]. A general result can then be stated in the following theorem.

Theorem 15.1. *If there exist a positive definite function $V(x)$ ($V(x) > 0$, $\forall x \neq 0$), a gain $\mathbb{K}$, a positive definite diagonal matrix M, a matrix $\mathbb{G}$ and two positive scalars ζ and δ_1 satisfying, for any admissible z and $i = 1,2,3$,*

$$\begin{aligned} \tfrac{\partial V}{\partial x}[(\mathbb{A}(z,x) + \mathbb{B}_1\mathbb{K})x + \mathbb{B}_1\phi(\mathbb{K}x) + \mathbb{B}_2(z,x)\omega] - 2\phi(\mathbb{K}x)'M(\phi(\mathbb{K}x) + \mathbb{G}x) \\ -\omega'\omega < 0, \end{aligned} \tag{15.32}$$

$$V(x) - x'(\mathbb{K}_{(i)} - \mathbb{G}_{(i)})' \frac{\frac{1}{\zeta} + \frac{1}{\delta_1}}{u_{0(i)}^2} (\mathbb{K}_{(i)} - \mathbb{G}_{(i)})x \geq 0, \tag{15.33}$$

$$V(x) - x'\mathbb{R}'_{(i)} \frac{\frac{1}{\zeta} + \frac{1}{\delta_1}}{\beta^2} \mathbb{R}_{(i)}x \geq 0, \tag{15.34}$$

$$V(x) - x'\mathbb{C}'_{(i)} \frac{\frac{1}{\zeta} + \frac{1}{\delta_1}}{u_{1(i)}^2} \mathbb{C}_{(i)}x \geq 0, \tag{15.35}$$

then the gain $\mathbb{K}$ and the sets $\mathscr{S}_1(V,\zeta,\delta_1) = \{x \in \Re^6; V(x) \leq \frac{1}{\zeta} + \frac{1}{\delta_1}\}$ and $\mathscr{S}_0(V,\zeta) = \{x \in \Re^6; V(x) \leq \zeta^{-1}\}$ are solutions to Problem 15.1.

Proof. The satisfaction of (15.33) means that the set $\mathscr{S}_1(V,\zeta,\delta_1)$ is included in the set $S(u_0)$ defined as in (15.22). Thus, one can conclude that for any $x \in \mathscr{S}_1(V,\zeta,\delta_1)$ the nonlinearity $\phi(\mathbb{K}x)$ satisfies the sector condition (15.23). Moreover, the satisfaction of relations (15.34) and (15.35) means that the set $\mathscr{S}_1(V,\zeta,\delta_1)$ is included in the set $\Omega(x)$ defined by (15.19). Thus, for any $x \in \mathscr{S}_1(V,\zeta,\delta_1)$, the constraints C2 and C3 are respected. Consider a positive definite function $V(x)$ ($V(x) > 0, \forall x \neq 0$). We want to prove that the time-derivative of this function satisfies (15.29) along the trajectories of the closed-loop system (15.21) for all admissible nonlinearity $\phi(\mathbb{K}x)$, all admissible uncertain vector z and all disturbances satisfying (15.11). Hence, using Lemma 15.1, one gets

$$\dot{V}(x) - \omega'\omega \leq \dot{V}(x) - 2\phi(\mathbb{K}x)'(\phi(\mathbb{K}x) + \mathbb{G}x) - \omega'\omega.$$

Thus, the satisfaction of relation (15.32) implies that

$$\dot{V}(x) - 2\phi(\mathbb{K}x)'M(\phi(\mathbb{K}x) + \mathbb{G}x) - \omega'\omega < 0$$

along the trajectories of system (15.21). Relations (15.30) and (15.31) are then satisfied. Therefore, when $\omega = 0$, one gets $\dot{V}(x) < 0, \forall x(0) \in \mathscr{S}_1(V,\zeta,\delta_1)$. When $\omega \neq 0$, the closed-loop trajectories of system (15.21) remain bounded in $\mathscr{S}_1(V,\zeta,\delta_1)$, for any $x(0) \in \mathscr{S}_0(V,\zeta)$ and any disturbance satisfying (15.11). One can conclude that the gain $\mathbb{K}$ and the sets $\mathscr{S}_1(V,\zeta,\delta_1) = \{x \in \Re^6; V(x) \leq \frac{1}{\zeta} + \frac{1}{\delta_1}\}$ and $\mathscr{S}_0(V,\zeta) = \{x \in \Re^6; V(x) \leq \zeta^{-1}\}$ are solutions to Problem 15.1. □

Theorem 15.1 provides a sufficient condition to solve the control gain design. However, such a condition appears not really constructive to exhibit a suitable function $V(x)$, a gain $\mathbb{K}$ and two scalars ζ and δ_1. The idea then consists in considering a quadratic function for $V(x)$, as $V(x) = x'Px$, $P = P' > 0$. Thus, by exploiting the description of the closed-loop system through a polytopic model (15.28), the following proposition derived from Theorem 15.1 can be stated.

Proposition 15.1. *If there exist symmetric positive definite matrices $W \in \Re^{6\times 6}$, $R_1 \in \Re^{3\times 3}$, a diagonal positive matrix $S \in \Re^{3\times 3}$, two matrices $Y \in \Re^{3\times 6}$ and $Z \in \Re^{3\times 6}$, three positive scalars ε, ζ and δ_1 satisfying*[1]

$$\begin{bmatrix}
W\mathbb{A}'_{1j} + \mathbb{A}_{1j}W + \mathbb{B}_1Y + Y'\mathbb{B}'_1 + \mathbb{R}'R_1\mathbb{R} \\
\quad +\varepsilon(u^2_{1(3)}(1+\beta^2)+\beta^2)\mathbb{R}'\mathbb{R} & \star & \star & \star & \star & \star \\
u_{1(2)}B_{2j}\mathbb{R}W & -R_1 & \star & \star & \star & \star \\
\begin{bmatrix} B_3 \\ I_3 \end{bmatrix}\mathbb{R}W & 0 & -\varepsilon I_6 & \star & \star & \star \\
S\mathbb{B}' - Z & 0 & 0 & -2S & \star & \star \\
B'_{4j}\mathbb{R} & 0 & 0 & 0 & -I_4 & \star \\
0 & 0 & 0 & 0 & B_{5j} & -\varepsilon I_3
\end{bmatrix} < 0, \tag{15.36}$$

[1] $\star$ stands for symmetric blocks.

$$\begin{bmatrix} W & \star & \star \\ Y_{(i)} - Z_{(i)} & \zeta u_{0(i)}^2 & \star \\ Y_{(i)} - Z_{(i)} & 0 & \delta_1 u_{0(i)}^2 \end{bmatrix} \geq 0, \tag{15.37}$$

$$\begin{bmatrix} W & \star & \star \\ \mathbb{R}_{(i)} W & \zeta \beta^2 & \star \\ \mathbb{R}_{(i)} W & 0 & \delta_1 \beta^2 \end{bmatrix} \geq 0, \tag{15.38}$$

$$\begin{bmatrix} W & \star & \star \\ \mathbb{C}_{(i)} W & \zeta u_{1(i)}^2 & \star \\ \mathbb{C}_{(i)} W & 0 & \delta_1 u_{1(i)}^2 \end{bmatrix} \geq 0, \tag{15.39}$$

$\forall i = 1,2,3$, *then, the control gain* $\mathbb{K} \in \Re^{3\times 6}$ *given by* $\mathbb{K} = YW^{-1}$ *is such that:*

(i) when $\omega \neq 0$*, the trajectories of the closed-loop system (15.18) remain bounded in the set*

$$\mathscr{E}_1(W,\zeta,\delta_1) = \left\{ x \in \Re^6; x'W^{-1}x \leq \frac{1}{\zeta} + \frac{1}{\delta_1} \right\}, \tag{15.40}$$

for any $x(0) \in \mathscr{E}_0(W,\zeta)$*,*

$$\mathscr{E}_0(W,\zeta) = \left\{ x \in \Re^6; x'W^{-1}x \leq \frac{1}{\zeta} \right\}, \tag{15.41}$$

and for any ω *satisfying (15.11);*

(ii) when $\omega = 0$*, the set* $\mathscr{E}_0(W,\zeta) = \mathscr{E}_1(W,\zeta,\delta_1)$ *is included in the basin of attraction of the closed-loop system (15.18) and is contractive.*

Proof. The proof mimics the one of Theorem 15.1. The satisfaction of relation (15.37) means that the set $\mathscr{E}_1(W,\zeta,\delta_1)$ defined in (15.40) is included in the set $S(u_0)$ defined in (15.22). Thus, one can conclude that for any $x \in \mathscr{E}_1(W,\zeta,\delta_1)$ the nonlinearity $\phi(\mathbb{K}x)$ satisfies the sector condition (15.23) with $\mathbb{G} = ZW^{-1}$. Furthermore, the satisfaction of relations (15.38) and (15.39) implies that the set $\mathscr{E}_1(W,\zeta,\delta_1)$ is included in the set $\Omega(x)$ defined in (15.19). Hence, for any admissible uncertain vector z (see (15.27)) and any admissible vector belonging to $\mathscr{E}_1(W,\zeta,\delta_1)$, the closed-loop system (15.21) or (15.24) can be written through the polytopic model (15.28).

The time-derivative of $V(x) = x'W^{-1}x$ along the trajectories of system (15.28) writes:

$$\begin{aligned} \dot{V}(x) - \omega'\omega \leq 2x'W^{-1} \sum_{j=1}^{4} \lambda_j \{ & [\mathbb{A}_{1j} + \mathbb{R}'T_{(2)}B_{2j}\mathbb{R} + \mathbb{R}'T_{(3)}(B_3 + D(e))\mathbb{R} + \mathbb{B}_1 YW^{-1}]x \\ & + \mathbb{R}'[B_{4j} + D(e)B_{5j}]\omega \} + 2x'W^{-1}\mathbb{B}_1\phi(\mathbb{K}x) \\ & - 2\phi(\mathbb{K}x)'M(\phi(\mathbb{K}x) + ZW^{-1}x) - \omega'\omega. \end{aligned}$$

By convexity one can prove that the right term of the above inequality is negative definite if

$$2x'W^{-1}[\mathbb{A}_{1j}+\mathbb{R}'T_{(2)}B_{2j}\mathbb{R}+\mathbb{R}'(B_3+D(e))\mathbb{R}+\mathbb{B}_1YW^{-1}]x+2x'W^{-1}\mathbb{B}_1\phi(\mathbb{K}x)$$
$$+2x'W^{-1}\mathbb{R}'(B_{4j}+D(e)B_{5j})\omega-2\phi(\mathbb{K}x)'M(\phi(\mathbb{K}x)+ZW^{-1}x)-\omega'\omega<0.$$

Thus, by using (15.12) one can upper-bound the term containing $T_{(2)}$ as [17]

$$2x'W^{-1}\mathbb{R}'T_{(2)}B_{2j}\mathbb{R}x\leq x'(W^{-1}\mathbb{R}'R_1\mathbb{R}W^{-1}+u_{1(2)}^2\mathbb{R}'B_{2j}'R_1^{-1}B_{2j}\mathbb{R})x$$

with $R_1=R_1'>0$. In the same way, by using both (15.12) and (15.19), one can upper-bound the term containing $T_{(3)}$ and $D(e)$ as

$$\begin{aligned}
&2x'W^{-1}\mathbb{R}'(T_{(3)}(B_3+D(e))\mathbb{R}x+D(e)B_{5j}\omega)=\\
&2x'W^{-1}\mathbb{R}'\left[\begin{array}{ccc}T_{(3)}I_3 & T_{(3)}D(e) & D(e)\end{array}\right]\left[\begin{array}{ccc}B_3\mathbb{R} & 0\\ \mathbb{R} & 0\\ 0 & B_{5j}\end{array}\right]\left[\begin{array}{c}x\\ \omega\end{array}\right]\\
&\leq\varepsilon(u_{1(3)}^2(1+\beta^2)+\beta^2)x'W^{-1}\mathbb{R}'\mathbb{R}W^{-1}x\\
&+\varepsilon^{-1}\left[\begin{array}{cc}x' & \omega'\end{array}\right]\left[\begin{array}{ccc}B_3\mathbb{R} & 0\\ \mathbb{R} & 0\\ 0 & B_{5j}\end{array}\right]'\left[\begin{array}{ccc}B_3\mathbb{R} & 0\\ \mathbb{R} & 0\\ 0 & B_{5j}\end{array}\right]\left[\begin{array}{c}x\\ \omega\end{array}\right].
\end{aligned}$$

Thus, the satisfaction of relations (15.36) to (15.39) with $M=S^{-1}$, $\mathbb{K}=YW^{-1}$ and $\mathbb{G}=ZW^{-1}$ allows to verify that

$$\dot{V}(x)<\omega'\omega, \tag{15.42}$$

for all $x(0)\in\mathscr{E}_0$ and any ω satisfying (15.11). Hence the trajectories of the closed-loop system (15.18) remain bounded in $\mathscr{E}_1(W,\zeta,\delta_1)$ for all $x(0)\in\mathscr{E}_0(W,\zeta)$ and any ω satisfying (15.11). This completes the proof of statement (i). To prove statement (ii), assume that $\omega=0$. Then, from (15.42), we have $\dot{V}(x)<0$, $\forall x\in\mathscr{E}_1$, which means that $\mathscr{E}_0=\mathscr{E}_1$ is a set of asymptotic stability for system (15.21). □

Remark 15.1. The study of system (15.18) subject to constraints (15.19) means that the constraints on the error (C2) and on the velocity (part 1 of C3) are linearly respected (saturation avoidance case). On the contrary, saturation of the acceleration (part 2 of C3) is allowed. Nevertheless, if one wants to consider that saturation on the velocity is also allowed then one can modify the closed-loop system as

$$\begin{aligned}
\dot{e} &= L(z,e)sat_{u_1}(u)+B(z,e)\omega,\\
\dot{u} &= sat_{u_0}(K_1e+K_2sat_{u_1}(u)),\\
T &= sat_{u_1}(u).
\end{aligned} \tag{15.43}$$

Thus, by considering $x_n=\left[\begin{array}{cc}e' & u'\end{array}\right]'\in\Re^6$ and the same type of matrices as in (15.15), the closed-loop system reads

$$\dot{x}_n=(\mathbb{A}(z,x_n)+\mathbb{B}_1\mathbb{K})x_n+\mathbb{B}_1\phi_0+(\mathbb{A}(z,x_n)+\mathbb{B}_1\mathbb{K})\mathbb{B}_1\phi_1+\mathbb{B}_2\omega, \tag{15.44}$$

where $\phi_0=sat_{u_0}(K_1e+K_2sat_{u_1}(u))-(K_1e+K_2sat_{u_1}(u))=sat_{u_0}(\mathbb{K}x_n+K_2\phi_1)-(\mathbb{K}x_n+K_2\phi_1)$ and $\phi_1=sat_{u_1}(u)-u=sat_{u_1}(\mathbb{C}x_n)-\mathbb{C}x_n$. In this case, Problem 15.1

can be studied with respect to system (15.44) by using nested dead-zone nonlinearities as in [20].

Remark 15.2. The disturbance free case (*i.e.*, $\omega = 0$) corresponds to the case of a fix target. In this case, the description of the closed-loop system is simpler. Furthermore, the condition C4 does not exist any more. Thus, the relations of Proposition 15.1 can be simplified, by removing roughly speaking all the lines and columns due to the presence of ω, and therefore are equivalent to those of Proposition 4 in [22].

15.4.2 Optimization Issues

It is important to note that relations (15.36), (15.37), (15.38) and (15.39) of Proposition 15.1 are LMIs. Depending on the energy bound on the disturbance, δ_1, is given by the designer or not, the following optimization problems can be considered:

- given δ_1, we want to optimize the size of the sets $\mathscr{E}_0$ and $\mathscr{E}_1$. This case can be addressed by the following convex optimization problem:

$$\begin{array}{c} \min\limits_{W,R_1,Y,Z,S,\varepsilon,\zeta,\delta,\sigma} \zeta+\delta+\sigma \\ \text{subject to} \\ \text{relations } (15.36),(15.37),(15.38),(15.39), \\ \begin{bmatrix} \sigma I_6 & \star \\ Y & I_3 \end{bmatrix} \geq 0, \begin{bmatrix} \delta I_6 & \star \\ I_6 & W \end{bmatrix} \geq 0. \end{array} \tag{15.45}$$

  The last two constraints are added to guarantee a satisfactory conditioning number for matrices $\mathbb{K}$ and W; and
- δ_1 being a decision variable, we want to minimize it. This problem comes to find the largest disturbance tolerance. In this case, we can add δ_1 in the previous criteria; the other decision variables may be kept in order to satisfy a certain trade-off between the size of the sets $\mathscr{E}_0$, $\mathscr{E}_1$ and δ_1.

15.5 Application

This section presents two applications of the proposed visual servo control approach for driving a wheeled robot equipped with a camera mounted on a pan-platform (Figure 15.2).

With respect to the world frame $R(O,X,Y,Z)$, x and y are the coordinates of the robot reference point M located at mid-distance between the wheels. Let $R_M(M,X_M,Y_M,Z_M)$ be a frame attached to the robot, with X_M directed along the robot main axis, and $R_P(P,X_P,Y_P,Z_P)$ a frame attached to the pan-platform, its origin P being located at its center of rotation P. Let θ denote the angle between X_M and the X-axis, and θ_p the angle between X_P and the X_M. The camera is rigidly fixed to the pan-platform. The transformation between R_P and the camera frame R_C consists of an horizontal translation of vector $\begin{bmatrix} a & b & 0 \end{bmatrix}'$ and a rotation of angle $\frac{\pi}{2}$ about the Y_P-axis. D_x is the distance between M and P. The velocity of the robot,

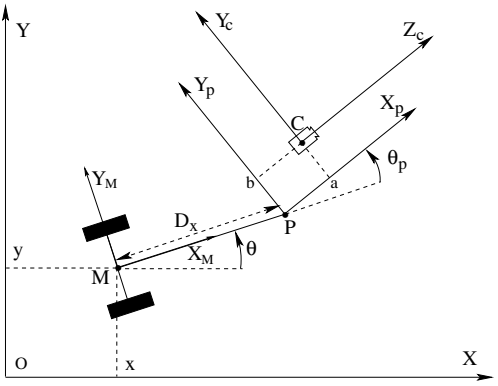


Fig. 15.2 Camera fixed on a pan-platform supported by a wheeled robot

which constitutes the actual system input is described by the vector $\dot{q} = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}'$, where v_1 and v_2 are the linear and the angular velocities of the cart with respect to R, respectively, while v_3 is the pan-platform angular velocity with respect to the robot main axis X_M. The system kinematics is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\theta}_p \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & 0 \\ \sin\theta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}.$$

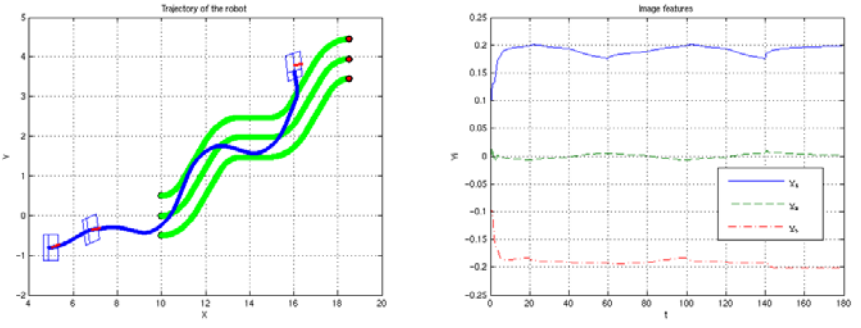
The kinematic screw of the camera T is linked to the velocity vector $\dot{q}$ by the relation $T = J(q)\dot{q}$, where $J(q)$ is the robot Jacobian given by

$$J(q) = \begin{bmatrix} -\sin\theta_p & D_x\cos\theta_p + a & a \\ \cos\theta_p & D_x\sin\theta_p - b & -b \\ 0 & -1 & -1 \end{bmatrix}. \tag{15.46}$$

As shown in [18], the rotational degree of freedom of the pan-platform allows to overcome the nonholonomic constraint of the wheeled base.

15.5.0.1 Application 1

In the first application, the robot has to track a moving target and stabilize its camera in front of it when it stops. The initial robot configuration is $\begin{bmatrix} x & y & \theta & \theta_p \end{bmatrix}' = \begin{bmatrix} 4.85m & -0.8m & 0rad & 0.175rad \end{bmatrix}'$, and the initial value of the state vector is $\xi(0) = \begin{bmatrix} -0.12m & -0.0159m & 0.0849m & 0ms^{-1} & 0ms^{-1} & 0rads^{-1} \end{bmatrix}'$. At the beginning of the task, the coordinates of the three target points, with respect to R, are: $E_1 \begin{bmatrix} 10m & 0.5m \end{bmatrix}'$, $E_2 \begin{bmatrix} 10m & 0m \end{bmatrix}'$ and $E_3 \begin{bmatrix} 10m & -0.5m \end{bmatrix}'$. According to condition C4, the target velocity is supposed to be unknown but square integrable, and its bound is defined by taking $\delta_1 = 12.1949$ in (15.11). The interval of distance between the robot and the



(a) (b)

Fig. 15.3 (a) Trajectory of the robot; and (b) visual features evolution

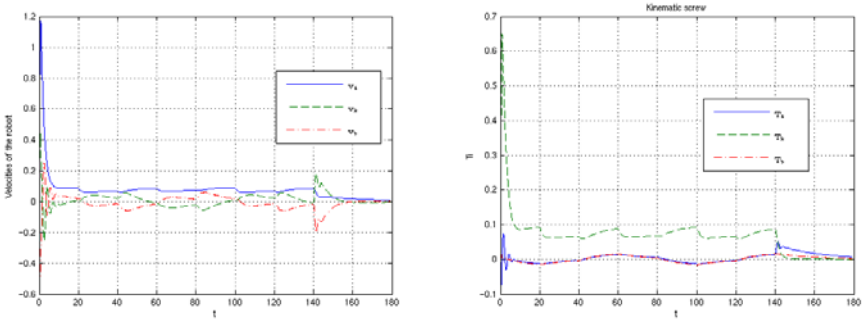
camera is given by taking $d_{min} = 2.454m$ and $d_{max} = 8m$ in (15.1). At the expected final position, the visual features in the camera image plane must reach the values: $Y_1^* = 0.2$, $Y_2^* = 0$, and $Y_3^* = -0.2$. To guarantee the visibility of the target, the coefficient β was fixed to 0.4. The bounds on the camera velocity and acceleration are $u_1 = \begin{bmatrix} 1 & 1 & 0.1 \end{bmatrix}'$, and $u_0 = \begin{bmatrix} 1 & 1 & 5 \end{bmatrix}'$. By applying the proposed control scheme with the MATLAB® LMI Control Toolbox we obtained the following value for the control gain $\mathbb{K}$:

$$\mathbb{K} = \begin{bmatrix} -505 & 1392.8 & -507.8 & -6.5 & 0.1 & 8.6 \\ -814 & 35.1 & 784.8 & 0.1 & -12.5 & 0.1 \\ -937.5 & -161.2 & -973.5 & 8.6 & 0.1 & -72.8 \end{bmatrix}.$$

Figure 15.3 represents the robot trajectory and the evolution of the visual features. As a result, the target is correctly tracked by the robot despite the uncertainties on the target-points depth and the unknown value of the target velocity. Finally, when the target stops, the robot succeeds in stabilizing the camera in front of it, at the expected relative configuration. The velocities of the robot and the kinematic screw of the camera are described in Figure 15.4. One can see in Figure 15.5 that the second components saturate during the very beginning of the task.

15.5.0.2 Application 2

The second application concerns the sequencing of two navigation tasks. The objective is to show the interest of the control approach to guarantee the satisfaction of constraints during the transition between tasks, whereas the switching of task-function may induce a strong variation of velocity. In the first task, the robot has to follow a wall by using its proximetric sensors and odometry. The second task is a visual servoing tasks which consists in driving the robot in order to position its camera in front of a fixed target.



(a) (b)

Fig. 15.4 (a) Robot velocities; and (b) camera kinematic screw T

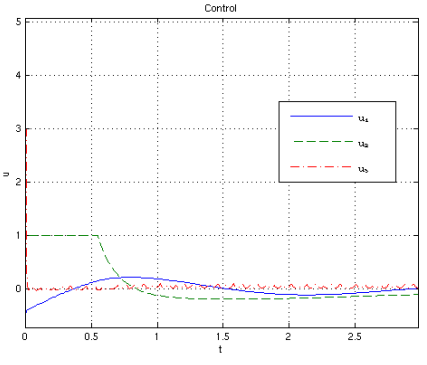


Fig. 15.5 Description of the control components during the very beginning of the task

Task 1. During the first task, the robot must follow the wall with a prescribed linear velocity v_r, while its on-board camera must be controlled in order to keep the target in its line of sight. Three kinds of sensory data are simulated. Odometry is used to regulate the robot linear velocity v_1; ultrasounds are used to detect the distance to the wall and the relative robot direction in order to control its angular velocity v_2; vision is used to control the camera angular velocity v_3. The task function $e_1(q,t)$ is defined by

$$e_1(q,t) = \begin{bmatrix} s_c - v_r t \\ y_e + \gamma \theta_e \\ Y_2 \end{bmatrix}, \tag{15.47}$$

where s_c is the arclength abscissa of the robot trajectory, v_r is the robot reference linear velocity, y_e is the distance between the point M and the wall, θ_e is the direction of the robot with respect to the wall, Y_2 is the ordinate of the projection of the target center in the camera image-plane and z_2 is its depth with respect to R_C; $\gamma > 0$ is a constant. The time-derivative of this task function reads

$$\dot{e}_1(q,t) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -\gamma & 0 \\ \multicolumn{3}{c}{L_{Y_2} J} \end{bmatrix}}_{H} \dot{q}(t) + \underbrace{\begin{bmatrix} -v_r \\ v_r \sin\theta_e \\ 0 \end{bmatrix}}_{B}, \tag{15.48}$$

where $L_{Y_2} = \begin{bmatrix} -1/z_2 & Y_2/z_2 & 1+Y_2^2 \end{bmatrix}$ is the interaction matrix associated with Y_2, and J is the Jacobian matrix given by (15.46). For this first task a kinematic controller is defined by imposing an exponential decay of the task function, as it is classically done for sensor-based control in robotics: $\dot{e}_1 = -\lambda e_1$. The following expression of $\dot{q}$ is deduced:

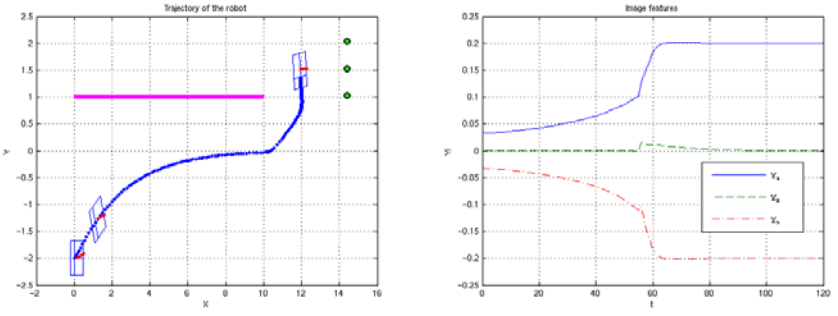
$$\dot{q} = -\lambda H^{-1}(e_1 + B). \tag{15.49}$$

Task 2. The second task is the visual servo control task described in Section 15.2.2 but with a fixed target (see also Proposition 4 in [22]). The multicriteria controller $\dot{T} = sat_{u_0}(\mathbb{K}\xi)$ is deduced from Proposition 15.1. The expression of the robot velocity vector $\dot{q}$ that constitutes the actual control input of the robot is then deduced from the value of T and the robot Jacobian (15.46). In order to smooth the transition between the two tasks, the approach proposed in [19] is followed. Each task e_i, $i = 1,2$, is valid within a region W_i: W_1 is a neighborhood of the wall inside which the proximetric data can be measured, and W_2 is the angular sector centered at the visual target position, which is defined by (15.1) and (15.2). Once the robot has reached the region $W_1 \cap W_2$ the task function $e_2(q,t)$ and its time derivative are evaluated, while the robot still executes the first task. At the switching time t_s the vision-based control loop is activated by considering the initial conditions $e_2(q,t_s)$ and $\dot{e}_2(q,t_s)$.

Simulation results. At the beginning of task 1 the robot's configuration is given by $x = 0$ m, $y = -2$ m, $\theta_0 = 0$ rad, and $\theta_p = 0.24$ rad. The reference wall is defined by the line $y = 1$ m with respect to frame R. As we impose the security distance of 1 m the actual reference path is the line $y = 0$. The parameters of the second task are as follows: The coordinates of the target point are $E_1 \begin{bmatrix} 14.42 \text{ m} & 2.033 \text{ m} \end{bmatrix}'$, $E_2 \begin{bmatrix} 14.42 \text{ m} & 1.533 \text{ m} \end{bmatrix}'$ and $E_3 \begin{bmatrix} 14.42 \text{ m} & 1.033 \text{ m} \end{bmatrix}'$ with respect to R. We fixed $d_{min} = 2.454$ m, and $d_{max} = 10$ m. The reference values for the projected target points are: $Y_1^* = 0.2$ m, $Y_2^* = 0$ m, and $Y_3^* = -0.2$ m. To ensure target visibility we consider $\beta = 0.4$. The bounds on the kinematic screw and the acceleration of the camera are: $u_1 = \begin{bmatrix} 1 \text{ m/s} & 1 \text{ m/s} & 10 \text{ rad/s} \end{bmatrix}'$, and $u_0 = \begin{bmatrix} 1 \text{ m/s}^2 & 1 \text{ m/s}^2 & 5 \text{ rad/s}^2 \end{bmatrix}'$. The switching time t_s was fixed to 55 s. By applying the proposed control scheme with the MATLAB® LMI Control Toolbox we obtained the following values of the control gain $\mathbb{K}$:

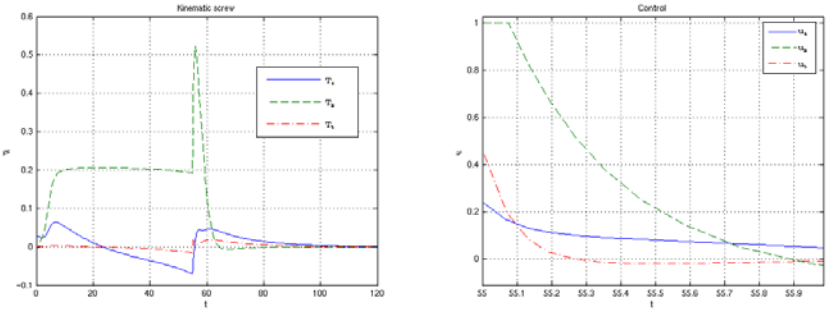
$$\mathbb{K} = \begin{bmatrix} -9.4931 & 27.1505 & -9.4931 & -4.7075 & 0 & -4.3744 \\ -19.6803 & 0 & -19.6803 & 0 & -6.2776 & 0 \\ -29.4251 & 55.1338 & -29.4251 & -4.4403 & 0 & -21.6589 \end{bmatrix}.$$

Figure 15.6 represents the robot trajectory and the convergence of the visual data Y_i to their reference value Y_i^*. Figure 15.7 represents the camera's kinematic screw along the motion and a zoom on the vision-based controller during the beginning



(a) (b)

Fig. 15.6 (a) Robot trajectory; and (b) visual feature evolution



(a) (b)

Fig. 15.7 (a) Kinematics screw evolution; and (b) zoom on the control

of the second task. The components of the camera's kinematics screw vary strongly at the transition but remain bounded. This condition ensures that the velocity vector $q(t)$ remains continuously differentiable during the complete motion. Now, considering the right picture of Figure 15.7, one can observe that the control components remain bounded, insuring that the limitation on the acceleration of the camera is satisfied. Such a guarantee could not be obtained with a simple second order dynamics as proposed in [19]. As it can be seen in Figure 15.7 (right picture), one of the control component saturates for a while at the beginning of the task. However, as this saturating condition has been considered in the control synthesis, the stability of the closed-loop system remains guaranteed.

15.6 Concluding Remarks and Perspectives

Techniques that allow to design a multicriteria image-based controller in order to track moving targets with square integrable velocity were proposed in this chapter.

Based on both polytopic and norm-bounded representations of uncertainties, and a modified sector condition for the description of saturation terms, the proposed controller stabilizes the camera despite unknown value of the target points depth, bounds on admissible visual features errors to ensure visibility, and limits on the camera velocity and acceleration. The main results are given as constructive conditions by exploiting Lyapunov analysis and LMI-based optimization procedures.

When dealing with such a problematic, there are still open problems. Several steps could be investigated: to use Lyapunov functions with more complex structures, for example as in [8]; to search for different structures of the control law, inspired for example by those issued from [23].

References

[1] Bellot, D., Danès, P.: Handling visual servoing schemes through rational systems and LMIs. In: IEEE Conference on Decision and Control (2001)

[2] Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

[3] Castelan, E.B., Tarbouriech, S., Gomes da Silva Jr., J.M., Queinnec, I.: $\mathscr{L}_2$-stabilization of continuous-time systems with saturating actuators. International Journal of Robust and Nonlinear Control 16, 935–944 (2006)

[4] Chaumette, F., Hutchinson, S.: Visual servo control part II: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)

[5] Chesi, G.: Visual servoing path-planning via homogeneous forms and LMI optimizations. IEEE Transactions on Robotics 25(2), 281–291 (2009)

[6] Chesi, G., Hashimoto, K., Prattichizzo, D., Vicino, A.: Keeping features in the field of view in eye-in-hand visual servoing: a switching approach. IEEE Transactions on Robotics 20(5), 908–913 (2004)

[7] Chesi, G., Vicino, A.: Visual servoing for large camera displacements. IEEE Transactions on Robotics 20(4), 724–735 (2004)

[8] Coutinho, D.F., Gomes da Silva Jr., J.M.: Estimating the region of attraction of nonlinear control systems with saturating actuators. In: American Control Conference (2007)

[9] Cuvillon, L., Laroche, E., Gangloff, J., de Mathelin, M.: GPC versus H1 control for fast visual servoing of medical manipulator including exibilities. In: IEEE Conference on Decision and Control (2005)

[10] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Transactions on Robotics and Automation 8(3), 313–326 (1992)

[11] Gao, B.: Contribution to multicriteria 2d image-based controller design (in french). Ph.D. thesis, University Paul Sabatier (2006), `http://tel.archives-ouvertes.fr/tel-00119789`

[12] Gomes da Silva Jr., J.M., Tarbouriech, S.: Anti-windup design with guaranteed regions of stability: an LMI-based approach. IEEE Transactions on Automatic Control 50(1), 106–111 (2005)

[13] Hadj-Abdelkader, H., Mezouar, Y., Martinet, P., Chaumette, F.: Catadioptric visual servoing from 3d straight lines. IEEE Transactions on Robotics 24(3), 652–665 (2008)

[14] Malis, E., Rives, P.: Robustness of image-based visual servoing with respect to depth distribution errors. In: IEEE International Conference on Robotics and Automation (2003)

[15] Marchand, E., Chaumette, F.: Feature tracking for visual servoing purposes. In: Robotics and Autonomous Systems (2005)
[16] Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Transactions on Robotics and Automation 18(4), 534–549 (2002)
[17] Petersen, I.R.: A stabilisation algorithm for a class of uncertain linear systems. Systems and Control Letters 8(4), 351–356 (1987)
[18] Pissard-Gibollet, R., Rives, P.: Applying visual servoing techniques to control a mobile hand-eye system. In: IEEE International Conference on Robotics and Automation (1995)
[19] Souères, P., Cadenat, V., Djeddou, M.: Dynamical sequence of multi-sensor based task for mobile robots navigation. In: IFAC Symposium on Robot Control (2003)
[20] Tarbouriech, S., Prieur, C., Gomes da Silva Jr., J.M.: Stability analysis and stabilization of systems presenting nested saturations. IEEE Transactions on Automatic Control 51(8), 1364–1371 (2006)
[21] Tarbouriech, S., Souères, P.: Advanced control strategies for the visual servoing scheme. In: IFAC Symposium on Robot Control (2000)
[22] Tarbouriech, S., Souères, P., Gao, B.: A multicriteria image-based controller based on a mixed polytopic and norm-bounded representation. In: Joint IEEE Conference on Decision and Control and European Control Conference, CDC-ECC (2005)
[23] Valmorbida, G., Tarbouriech, S., Garcia, G.: State feedback design for input-saturating nonlinear quadratic systems. In: American Control Conference (2009)
[24] Zachi, A.R.L., Hsu, L., Lizarralde, F.: Performing stable 2D adaptive visual positionning/tracking control without explicit depth measurement. In: IEEE International Conference on Robotics and Automation (2004)

Chapter 16
Points-based Visual Servoing with Central Cameras

Hicham Hadj-Abdelkader, Youcef Mezouar, and Philippe Martinet

Abstract. This chapter concerns hybrid visual servoing schemes from a set of points viewed by central camera. The main purpose is to decouple the velocity commands in order to obtain an adequate camera trajectory. The proposed schemes are model-free since they are based on the homography matrix between two views. The rotational motions are controlled using the estimated orientation between the current and the desired positions of the robot, while the translational motions are controlled using the combination between image points (onto the sphere or into the normalized plane) and 3D information extracted from the homography matrix. Real-time experimental results with a cartesian manipulator robot are presented and show clearly the decoupling properties of the proposed approaches.

16.1 Introduction

In vision-based control, the choice of the set of visual features to be used in the control scheme is still an open question, despite of the large quantity of results obtained in the last few years. The visual servoing schemes can be classified in three groups: position-based visual servoing (PBVS) [27], image-based visual servoing (IBVS) [7] and hybrid visual servoing [15]. In PBVS, the used information is defined in the 3D space which allow the control scheme to ensure nice decoupling properties between the degrees of freedom (DOF) (refer to [26]). Adequate 3D trajectories can thus be obtained such as a geodesic for the rotational motion and a straight line for the translational motion. However, this kind of control scheme is sensitive to measurement noises and the control may thus suffer from potential instabilities [3]. In IBVS the control is performed in the image space. Whatever the nature of the possible measures extracted from the image, the main question is how to combine them to obtain an adequate behavior of the system. In most works, the combination of different features is nothing but a simple stacking. If the error between the initial

Hicham Hadj-Abdelkader, Youcef Mezouar, and Philippe Martinet
LASMEA, University Blaise Pascal, Campus des Cezeaux, 63177 Aubiere, France
e-mail: {hadj,mezouar,martinet}@lasmea.univ-bpclermont.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 295–313.
springerlink.com

value of the features and the desired one is small, and if the task to realize constrains all the available DOF, that may be a good choice. However, as soon as the error is large, problems may appear such as reaching local minimum or task singularities [3]. Hybrid visual servoing is an alternative to the two previous control schemes. In this case, the visual features gather 2D and 3D information.

The way to design adequate visual features is directly linked to the modeling of their interaction with the robot motion, from which all control properties can be analyzed theoretically. If the interaction is too complex (*i.e.* highly nonlinear and coupled), the analysis becomes impossible and the behavior of the system is generally not satisfactory in difficult configurations where large displacements (especially rotational ones) have to be realized. To overcome these problems, it is possible to combine path-planning and visual servoing, since tracking planned trajectories allows the error to always remain small [20]. A second approach is to use the measures to build particular visual features that will ensure expected properties of the control scheme (refer for instance to [21, 14, 5, 13, 12, 4, 24]).

This chapter is concerned with homography-based visual servo control techniques with central catadioptric cameras. This framework, also called 2 1/2 D visual servoing [15] in the case where the image features are points, exploits a combination of reconstructed Euclidean information and image features in the control design. The 3D information is extracted from an homography matrix relating two views of a reference plane. As a consequence, the 2 1/2 D visual servoing scheme does not require any 3D model of the target. Unfortunately, in such approach when conventional cameras are used, the image of the target is not guaranteed to remain in the camera field of view. To overcome this deficiency, 2 1/2 D visual servoing is first extended to the entire class of central cameras (including pinhole cameras, central catadioptric cameras and some fisheye cameras [6]). It will be shown that as when a conventional camera is employed, the resulting interaction matrix is block-triangular with partial decoupling properties. Then two new control schemes will be proposed. The basic idea of the first one is to control the translational motions using a scaled 3D point directly obtained from the image points coordinates and the homography matrix. Compared to the conventional 2 1/2 D visual servoing, it allows to obtain better camera trajectory since the translation is controlled in the 3D space while the interaction matrix remains block-triangular. Then, a hybrid scheme which allow us to fully decouple rotational motions from translational ones (*i.e.* the resulting interaction matrix is square block-diagonal) will be proposed. For the three proposed control schemes, it will be also shown that the equilibrium point is globally stable even in the presence of errors in the norm of 3D points which appears in the interaction matrices.

16.2 Modeling

In this section, the unified cental projection model using the unitary sphere is briefly recalled. Then, Euclidean reconstruction from the generic homography matrix is addressed.

16.2.1 Generic Projection Model

Central imaging systems can be modeled using two consecutive projections: spherical projection succeeded by a perspective one. This geometric formulation called unified model has been proposed by Geyer and Daniilidis in [9] and has been intensively used by the vision and robotics community (structure from motion, calibration, visual servoing, *etc*).

Consider the virtual unitary sphere centered in the origin of the mirror frame $\mathscr{F}_m$ as shown in Figure 16.1 and the perspective camera centered in the origin of the camera frame $\mathscr{F}_c$. Without lost of generality, a simple translation of $-\xi$, along the Z axis of the mirror frame, between $\mathscr{F}_m$ and $\mathscr{F}_c$ is considered. Let $\mathscr{X}$ be a 3D point with coordinates $\mathbf{X} = [X\ Y\ Z]^\top$ in $\mathscr{F}_m$. The world point $\mathscr{X}$ is projected in the image plane into the point of homogeneous coordinates $\mathbf{x_i} = [x_i\ y_i\ 1]^\top$. The image formation process can be split in three steps given in the following:

- the 3D world point $\mathscr{X}$ is mapped onto the unit sphere surface:

$$\mathbf{X_s} = \frac{1}{\rho}\left[\mathrm{X\ Y\ Z}\right]^\top, \tag{16.1}$$

where $\rho = \|\mathbf{X}\| = \sqrt{X^2+Y^2+Z^2}$;

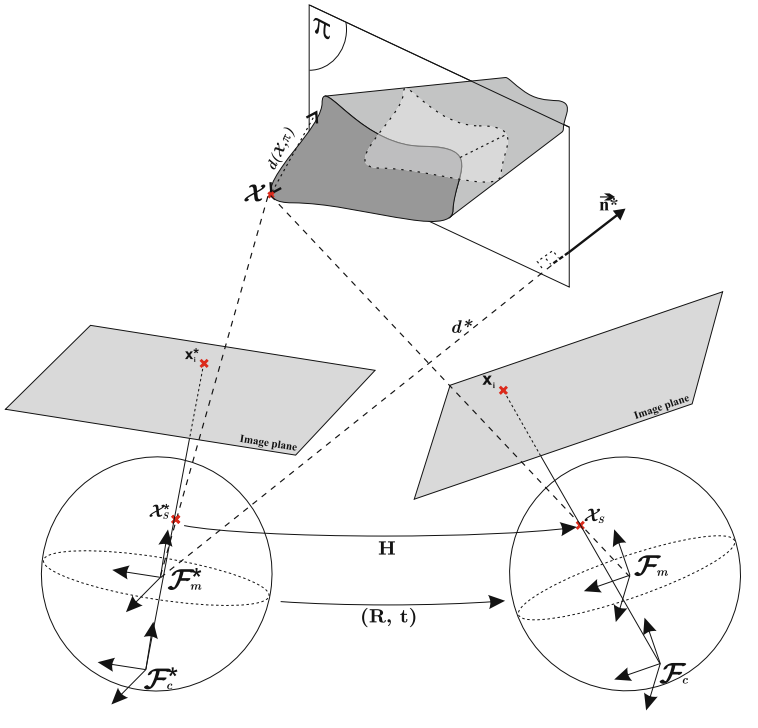


Fig. 16.1 Unified central projection and two views geometry

- the point $\mathbf{X_s}$ lying on the unitary sphere is perspectively projected on the normalized image plane $Z = 1 - \xi$ into a point of homogeneous coordinates:

$$\underline{\mathbf{x}} = \mathbf{f}(\mathbf{X}) = \left[\frac{X}{Z+\xi\rho} \;\; \frac{Y}{Z+\xi\rho} \;\; 1 \right]^\top \tag{16.2}$$

  (as it can be seen, the perspective projection model is obtained by setting $\xi = 0$); and
- the 2D projective point $\underline{\mathbf{x}}$ is mapped into the pixel image point with homogeneous coordinates $\underline{\mathbf{x_i}}$ using the collineation matrix $\mathbf{K}$:

$$\underline{\mathbf{x_i}} = \mathbf{K}\underline{\mathbf{x}}$$

  where the matrix $\mathbf{K}$ contains the conventional camera intrinsic parameters coupled with mirror intrinsic parameters, and can be written as

$$\mathbf{K} = \begin{bmatrix} f_u & \alpha_{uv} & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The matrix $\mathbf{K}$ and the parameter ξ can be obtained after calibration using for instance the method proposed in [1]. The inverse projection from the image plane onto the unit sphere can be obtained by inverting the second and last steps. As a matter of fact, the point $\underline{\mathbf{x}}$ in the normalized image plane is obtained using the inverse mapping $\mathbf{K^{-1}}$:

$$\underline{\mathbf{x}} = [x \; y \; 1]^\top = \mathbf{K^{-1}}\underline{\mathbf{x_i}}. \tag{16.3}$$

The point onto the unit sphere is then obtained by inverting the nonlinear projection (16.2):

$$\mathbf{X_s} = \mathbf{f^{-1}}(\underline{\mathbf{x}}) = \eta \left[x \; y \; 1 - \frac{\xi}{\eta} \right]^\top, \tag{16.4}$$

where

$$\eta = \frac{\xi + \sqrt{1 + (1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}.$$

16.2.2 Scaled Euclidean Reconstruction

Several methods were proposed to obtain the Euclidean reconstruction from two views [8]. They are generally based on the estimation of the essential or homography matrices. The epipolar geometry of cameras obeying the unified model has been recently investigated [10, 23, 11]. For control purposes, the methods based on the essential matrix are not well suited since degenerate configurations such as pure rotation motion can induce unstable behavior of the control scheme. It is thus preferable to use methods based on the homography matrix.

It will be shown now how one can compute the Homographic relationship between two central views of points. Consider two positions $\mathscr{F}_m$ and $\mathscr{F}_m^\star$ of the central camera (see Figure 16.1). Those frames are related by the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. Let (π) a 3D reference plane given in $\mathscr{F}_m^\star$ by the vector $\pi^{\star\top} = [\mathbf{n}^{\star\top} \; -d^\star]$, where $\mathbf{n}^\star$ is its unitary normal in $\mathscr{F}_m^\star$ and $d^\star$ is the distance from (π) to the origin of $\mathscr{F}_m^\star$.

Let $\mathscr{X}$ be a 3D point with coordinates $\mathbf{X} = [X \; Y \; Z]^\top$ with respect to $\mathscr{F}_m$ and with coordinates $\mathbf{X}^\star = [X^\star \; Y^\star \; Z^\star]^\top$ with respect to $\mathscr{F}_m^\star$. Its projection in the unit sphere for the two camera positions is given by the coordinates $\mathbf{X_s} = \rho^{-1}\mathbf{X}$ and $\mathbf{X_s^\star} = \rho^{\star-1}\mathbf{X}^\star$. The distance $d(\mathscr{X},\pi)$ from the world point $\mathscr{X}$ to the plane (π) is given by the scalar product $[\mathbf{X}^{\star\top} \; 1] \cdot \pi^\star$:

$$d(\mathscr{X},\pi) = \rho^\star \mathbf{n}^{\star\top}\mathbf{X_s^\star} - d^\star. \tag{16.5}$$

The relationship between the coordinates of $\mathscr{X}$ with respect to $\mathscr{F}_m$ and $\mathscr{F}_m^\star$ can be written as a function of their spherical coordinates:

$$\rho\mathbf{X_s} = \rho^\star\mathbf{R}\mathbf{X_s^\star} + \mathbf{t}. \tag{16.6}$$

By multiplying and dividing the translation vector by the distance $d^\star$ and according to (16.5), the expression (16.6) can be rewritten as

$$\rho\mathbf{X_s} = \rho^\star\mathbf{H}\mathbf{X_s^\star} + \alpha\mathbf{t}, \tag{16.7}$$

with $\mathbf{H} = \mathbf{R} + \dfrac{\mathbf{t}}{d^\star}\mathbf{n}^{\star\top}$ and $\alpha = -\dfrac{d(\mathscr{X},\pi)}{d^\star}$. $\mathbf{H}$ is the Euclidean homography matrix written as a function of the camera displacement and of the plane coordinates with respect to $\mathscr{F}_m^\star$. It has the same form as in the conventional perspective case (it can be decomposed into a rotation matrix and a rank 1 matrix). If the world point $\mathscr{X}$ belongs to the reference plane (π) (*i.e.* $\alpha = 0$) then (16.7) becomes

$$\mathbf{X_s} \propto \mathbf{H}\mathbf{X_s^\star}.$$

The homography matrix $\mathbf{H}$ related to the plane (π) can be estimated up to a scale factor by solving the linear equation $\mathbf{X_s} \otimes \mathbf{H}\mathbf{X_s^\star} = 0$ (where $\otimes$ denotes the cross-product) using, at least, four couples of coordinates $(\mathbf{X_s}_k; \mathbf{X_s^\star}_k)$ (where $k = 1 \cdots n$ with $n \geq 4$), corresponding to the spherical projection of world points $\mathscr{X}_k$ belonging to (π). If only three points belonging to (π) are available then at least five supplementary points are necessary to estimate the homography matrix by using for example the linear algorithm proposed in [15].

From the estimated homography matrix, the camera motion parameters (that is the rotation $\mathbf{R}$ and the scaled translation $\mathbf{t}_{d^\star} = \frac{1}{d^\star}\mathbf{t}$) and the structure of the observed scene (for example the vector $\mathbf{n}^\star$) can thus be determined (refer to [8, 28]). It can also be shown that the ratio $\sigma = \dfrac{\rho}{\rho^\star}$ can be computed as

$$\sigma = \frac{\rho}{\rho^\star} = \det(\mathbf{H}) \frac{\mathbf{n}^{\star\top} \mathbf{X}_\mathbf{s}^\star}{\mathbf{n}^{\star\top} \mathbf{R}^\top \mathbf{X}_\mathbf{s}}. \tag{16.8}$$

In the sequel, the rotation parameters and the ratio σ, extracted from the estimated homography are used to define the task function for the proposed hybrid visual servoing schemes.

16.3 Visual Servoing

In this section, the task function formalism and the interaction matrix for 2D points are concisely recalled. Then, three free-model hybrid visual servoing schemes with nice decoupling properties are described.

16.3.1 Task Function and Interaction Matrices

As usual when designing a visual servoing scheme, the visual feature vector $\mathbf{s}$ is often expressed as function of the 3D representation of the observed object such as a set of 3D points. In order to control the movements of the robotic system from visual features, one defines a task function to be regulated to $\mathbf{0}$ as [22]

$$\mathbf{e} = \mathbf{L}^+ (\mathbf{s} - \mathbf{s}^\star),$$

where $.^+$ denote the pseudo-inverse and $\mathbf{L}$ is the interaction matrix which links the variation of $\mathbf{s}$ to the camera velocities. If the observed object is motionless, one gets

$$\dot{\mathbf{s}} = \mathbf{L}\tau,$$

where τ is a 6D vector denoting the velocity screw of the central camera. The vector τ contains the instantaneous linear velocity $\mathbf{v}$ and the instantaneous angular velocity ω of the sensor frame expressed in the same frame. In the sequel, the sensor frame is chosen as the mirror frame $\mathscr{F}_m$.

A simple control law can be designed by imposing an exponential decay of the task function $\mathbf{e}$ toward $\mathbf{0}$:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e},$$

where λ is a proportional gain. The corresponding control law is

$$\tau = -\lambda \mathbf{L}^+ (\mathbf{s} - \mathbf{s}^\star). \tag{16.9}$$

In order to compute the control law (16.9), the interaction matrix $\mathbf{L}$ or its pseudo-inverse (its inverse if $\mathbf{L}$ is square) should be provided. In practice, an approximation $\widehat{\mathbf{L}}$ of the interaction matrix is used. If the task function $\mathbf{e}$ is correctly computed, the global asymptotic stability of the system can be obtained if the necessary and sufficient condition $\mathbf{L}\widehat{\mathbf{L}}^+ > 0$ is satisfied.

When the visual features are related to the projection of 3D points, the vector $\mathbf{s}$ is function of the 3D coordinates $\mathbf{X} = [X\ Y\ Z]^\top$ of the 3D point $\mathscr{X}$. In that case, the interaction matrix related to $\mathbf{s}$ can be written as

$$\mathbf{L} = \frac{\partial \mathbf{s}}{\partial \mathbf{X}} \mathbf{L}_{\mathbf{X}},$$

$\mathbf{J_s} = \dfrac{\partial \mathbf{s}}{\partial \mathbf{X}}$ is the Jacobian matrix linking the variations of $\mathbf{s}$ and $\mathbf{X}$, and $\mathbf{L_X}$ is the interaction matrix related to the 3D point $\mathscr{X}$:

$$\dot{\mathbf{X}} = \mathbf{L_X}\tau = \begin{pmatrix} -\mathbf{I_3} & [\mathbf{X}]_\times \end{pmatrix} \tau, \tag{16.10}$$

where $[\mathbf{a}]_\times$ is the anti-symmetric matrix of the vector $\mathbf{a}$.

If one considers n visual features related to the same 3D point $\mathscr{X}$, the global interaction matrix $\mathbf{L}$ for the features vector $\mathbf{s} = [\mathbf{s_1}\ \mathbf{s_2}\ \cdots\ \mathbf{s_n}]^\top$ can be written as

$$\mathbf{L} = \begin{pmatrix} \mathbf{J_{s_1}}^\top & \mathbf{J_{s_2}}^\top & \cdots & \mathbf{J_{s_n}}^\top \end{pmatrix}^\top \mathbf{L_X}.$$

16.3.2 Interaction Matrix for 2D Point

Consider a 3D point $\mathscr{X}$ with coordinates $\mathbf{X} = [X\ Y\ Z]^\top$ with respect to the mirror frame $\mathscr{F}_m$. Its central projection on the normalized image plane is obtained using (16.1) and it is given by the point of homogeneous coordinates $\underline{\mathbf{x}} = [x\ y\ 1]^\top$. If the visual feature is chosen as $\mathbf{s} = [x\ y]^\top$, the interaction matrix $\mathbf{L}$ is

$$\mathbf{L} = \mathbf{J_s}\, \mathbf{L_X},$$

where

$$\mathbf{J_s} = \frac{1}{\rho(Z+\xi\rho)^2} \begin{pmatrix} \rho Z + \xi(Y^2+Z^2) & -\xi XY & -X(\rho+\xi Z) \\ -\xi XY & \rho Z + \xi(X^2+Z^2) & -Y(\rho+\xi Z) \end{pmatrix}.$$

After few developments, the analytical expression of the interaction matrix $\mathbf{L}$ can be written as

$$\mathbf{L} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \end{pmatrix} \tag{16.11}$$

where

$$\mathbf{A} = \rho^{-1} \begin{pmatrix} -\frac{\gamma+\xi\left(x^2+y^2\right)}{1+\xi\gamma} + \xi x^2 & \xi xy & \gamma x \\ \xi xy & -\frac{\gamma+\xi\left(x^2+y^2\right)}{1+\xi\gamma} + \xi y^2 & \gamma y \end{pmatrix},$$

and

$$\mathbf{B} = \begin{pmatrix} xy & -\gamma\frac{\gamma+\xi\left(x^2+y^2\right)}{1+\xi\gamma} + y^2 & y \\ \gamma\frac{\gamma+\xi\left(x^2+y^2\right)}{1+\xi\gamma} - x^2 & -xy & -x \end{pmatrix}$$

with $\gamma = \sqrt{1+(1-\xi^2)(x^2+y^2)}$.

16.3.3 Decoupled Visual Servoing

In visual servoing scheme, the control properties are directly linked to the interaction between the designed features and the camera (or the robot) motion. The behavior of the camera depends on the coupling between the features and the camera velocities. For example, the interaction matrix in (16.11) related to the image coordinates of 2D points is highly nonlinear and coupled. Thereof, large displacements of the camera became difficult to realize.

Several approaches have been proposed to overcome these problems. Most of them ensure a good decoupling properties by combining 2D and 3D information when defining the input of the control law. The related control schemes are called hybrid visual servoing. In this work, three model free decoupled control schemes are proposed. Let us first define the observation vector as

$$\mathbf{s} = \left[\tilde{\mathbf{s}}^\top \quad \theta\mathbf{u}^\top \right]^\top.$$

The vector $\tilde{\mathbf{s}}$ is chosen to be variant to the translational motions of the camera and can be variant or invariant to the rotational motions, whereas the vector $\theta\mathbf{u}$, representing the rotational information between the current and the desired positions of the camera, is invariant to the translational motions. Consequently, the global interaction matrix $\mathbf{L}$ related to the features vector $\mathbf{s}$ is a block-triangular matrix:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{\tilde{\mathbf{s}}_v} & \mathbf{L}_{\tilde{\mathbf{s}}_\omega} \\ \mathbf{0_3} & \mathbf{L}_\omega \end{pmatrix}.$$

Note that when $\tilde{\mathbf{s}}$ is invariant to rotational motions, $\mathbf{L}$ becomes a block-diagonal matrix.

16.3.3.1 Interaction Matrix $\mathbf{L}_\omega$

The rotation matrix between the current and the desired positions of the central camera, can be obtained from the estimated homography matrix $\mathbf{H}$. Several representations of the rotation are possible. The representation $\theta\mathbf{u}$ (where θ is the rotation angle and $\mathbf{u}$ is a unit vector along the rotation axis) is chosen since it provides the largest possible domain for the rotation angle. The corresponding interaction matrix can be obtained from the time derivative of $\theta\mathbf{u}$ since it can be expressed with respect to the central camera velocity screw τ:

$$\frac{d(\theta\mathbf{u})}{dt} = \begin{pmatrix} \mathbf{0_3} & \mathbf{L}_\omega \end{pmatrix} \tau$$

where $\mathbf{L}_\omega$ is given by [17]

$$\mathbf{L}_\omega = \mathbf{I_3} - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{\operatorname{sinc}(\theta)}{\operatorname{sinc}^2(\frac{\theta}{2})}\right)[\mathbf{u}]_\times^2.$$

Note also that theoretically in this case $\mathbf{L}_{\omega}^{-1}\,\theta\mathbf{u} = \theta\mathbf{u}$. This nice property can advantageously be exploited to compute the control vector.

In practice, estimated camera parameters are used. The estimated rotation parameter $\widehat{\theta\mathbf{u}}$ can be written as a nonlinear function of the real ones $\psi(\theta\mathbf{u})$. Since $\widehat{\mathbf{L}_{\omega}^{-1}}\ \widehat{\theta\mathbf{u}} = \widehat{\theta\mathbf{u}}$, the closed-loop equation of the rotation control is

$$\frac{\theta\mathbf{u}}{dt} = -\lambda\mathbf{L}_{\omega}\psi(\theta\mathbf{u}).$$

The asymptotic stability of this system has been studied for conventional camera ($\xi = 0$) since in this case the function ψ has a simple analytical form [16]. However, the stability analysis remains an open problem when $\xi \neq 0$ since the nonlinear function ψ is much more complex in this case.

16.3.3.2 2 1/2 D Visual Servoing

2 1/2 D visual servoing has been first proposed by Malis and Chaumette in case of conventional camera ($\xi = 0$). In this section, the original scheme is extended to the entire class of central cameras. In order to control the translational motion, let us define $\tilde{\mathbf{s}}$ as

$$\tilde{\mathbf{s}} = \left[\,\tilde{\mathbf{s}}_{\mathbf{1}}{}^{\top}\ \ \tilde{\mathbf{s}}_{\mathbf{2}}\,\right]^{\top}$$

where $\tilde{\mathbf{s}}_{\mathbf{1}} = [x\ y]^{\top}$ and $\tilde{\mathbf{s}}_{\mathbf{2}} = \log(\rho)$ are respectively the coordinates of an image point and the logarithm of the norm of its corresponding 3D point. The error between the current value $\log(\rho)$ and the desired value $\log(\rho^{\star})$ can be estimated using (16.8) since $\tilde{\mathbf{s}}_{\mathbf{2}} - \tilde{\mathbf{s}}_{\mathbf{2}}{}^{\star} = \log(\sigma)$.

The corresponding interaction matrix $\mathbf{L}_{\tilde{\mathbf{s}}}$ can be written as

$$\mathbf{L}_{\tilde{\mathbf{s}}} = \left(\,\mathbf{J}_{\tilde{\mathbf{s}}_{\mathbf{1}}}{}^{\top}\,\mathbf{J}_{\tilde{\mathbf{s}}_{\mathbf{2}}}\,\right)^{\top}\mathbf{L}_{\mathbf{X}}$$

where the Jacobian matrix $\mathbf{J}_{\tilde{\mathbf{s}}_{\mathbf{1}}}$ is given by (16.11), and $\mathbf{J}_{\tilde{\mathbf{s}}_{\mathbf{2}}}$ can be easily computed:

$$\mathbf{J}_{\tilde{\mathbf{s}}_{\mathbf{2}}} = \rho^{-2}\,\mathbf{X}^{\top}.$$

$\mathbf{L}_{\tilde{\mathbf{s}}} = \left(\,\mathbf{L}_{\tilde{\mathbf{s}}_{\mathbf{v}}}\ \ \mathbf{L}_{\tilde{\mathbf{s}}_{\omega}}\,\right)$ can be obtained by stacking the interaction matrix in (16.11) and

$$\mathbf{L}_{\tilde{\mathbf{s}}_{\mathbf{2}}} = \mathbf{J}_{\tilde{\mathbf{s}}_{\mathbf{2}}}\mathbf{L}_{\mathbf{X}} = \frac{1}{\sigma\rho^{\star}}\left(\,-\Phi x\ \ -\Phi y\ \ \Phi\,\frac{\xi^2(x^2+y^2)-1}{1+\gamma\xi}\ \ 0\ \ 0\ \ 0\,\right), \tag{16.12}$$

with $\Phi = \frac{Z+\xi\rho}{\rho} = \frac{1+\gamma\xi}{\gamma+\xi(x^2+y^2)}$. Note that the parameter $\rho^{\star}$ can be estimated only once during an off-line learning stage. If the system is supposed correctly calibrated and that measurements are noiseless, then the control law is asymptotically stable for any positive value $\widehat{\rho^{\star}}$. However, the robustness with respect to calibration and measurement errors still remains an open problem.

16.3.3.3 Norm-ratio-based Visual Servoing

As it can be seen in (16.12), the ratio between ρ and $\rho^\star$ is invariant to rotational motion. In the sequel, this property will be exploited in a new control scheme allowing us to decouple translational motions from the rotational ones. At this end, let us now define $\tilde{\mathbf{s}}$ as

$$\tilde{\mathbf{s}} = \begin{bmatrix} \log(\rho_1) & \log(\rho_2) & \log(\rho_3) \end{bmatrix}^\top.$$

The interaction matrix $\mathbf{J}_{\tilde{\mathbf{s}}}$ corresponding to $\tilde{\mathbf{s}}$ is obtained by stacking the interaction matrices given by (16.12) for each point. In this case, the global interaction matrix $\mathbf{L}$ is a block-diagonal matrix:

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{\tilde{\mathbf{s}}_v} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{L}_\omega \end{pmatrix}.$$

As above-mentioned, the translational and rotational controls are fully decoupled. If the system is correctly calibrated and the measurements are noiseless, the system is stable since $\frac{\widehat{\rho^\star}_i}{\rho_i^\star}$ is positive.

16.3.3.4 Scaled 3D Point-based Visual Servoing

Visual servoing scheme based on 3D points benefits of nice decoupling properties [19] [2]. Recently, Tatsambon *et al.* show in [25] that similar decoupling properties than the ones obtained with 3D points can be obtained using visual features related to the spherical projection of a sphere: the 3D coordinates of the center of the sphere computed up to a scale (the inverse of the sphere radius). However, even if such an approach is theoretically attractive, it is limited by a major practical issue since spherical object has to be observed.

Consider a 3D point $\mathscr{X}$ with coordinates $\mathbf{X} = [X\ Y\ Z]^\top$ with respect to the frame $\mathscr{F}_m$. The corresponding point onto the unit sphere is $\mathbf{X_s}$ and $\mathbf{X} = \rho \mathbf{X_s}$.

Let us now choose $\tilde{\mathbf{s}}$ as

$$\tilde{\mathbf{s}} = \sigma\, \mathbf{X_s} = \frac{1}{\rho^\star}\, \mathbf{X}, \tag{16.13}$$

where $\rho^\star$ is the 2-norm of $\mathscr{X}$ with respect to the desired position $\mathscr{F}^\star$ of the camera. The feature vector $\tilde{\mathbf{s}}$ is thus defined as a vector containing the 3D point coordinates up to a constant scale factor. Its corresponding interaction matrix can be obtained directly from (16.10):

$$\mathbf{L}_{\tilde{\mathbf{s}}} = \frac{1}{\rho^\star}\mathbf{L}_{\mathbf{X}} = \begin{pmatrix} -\frac{1}{\rho^\star}\mathbf{I_3} & [\tilde{\mathbf{s}}]_\times \end{pmatrix}.$$

As it is shown in the expression of $\mathbf{L}_{\tilde{\mathbf{s}}}$, the only unknown parameter is $\rho^\star$ which appears as a gain on the translational velocities. A nonzero positive value attributed to $\rho^\star$ will thus ensure the global asymptotic stability of the control law. The ratio between the real value of $\rho^\star$ and the estimated one $\widehat{\rho^\star}$ will act as an over-gain in the translational velocities.

Note that a similar approach using conventional camera has been proposed by Malis and Chaumette in [16] in order to enhance the stability domain. However, an adaptive control law has to be used in order that the reference point remains in the camera field of view during the servoing task. This is not a crucial issue in our case since our approach can be used with a large field of view.

16.4 Results

The proposed hybrid visual servoing schemes have been validated with a series of experiments. They were carried out on a 6 DOF manipulator robot in eye-in-hand configuration. A fisheye camera is mounted on the end-effector of the robot (see Figure 16.2). The estimated camera calibration parameters are $\xi = 1.634$, $f_u = 695$, $f_v = 694.9$, $\alpha_{uv} = 0$, $u_0 = 400.4$ and $v_0 = 304.4$. In order to simplify the features extraction and tracking, the target is composed of a set of white marks printed into a black background. These marks are tracked and their centers of gravity are extracted using the VISP library [18]. The experiments are detailed in the sequel by denoting with:

- **A** the 2D point-based control law;
- **B** the hybrid scheme presented in Section 16.3.3.2;
- **C** the hybrid scheme presented in Section 16.3.3.3; and
- **D** the hybrid scheme presented in Section 16.3.3.4.

Experiment 1. A large generic displacement is considered. It is composed of a translation $\mathbf{t} = [80\ 80\ -40]$ cm and of a rotation $\theta\mathbf{u} = [0\ 50\ 140]$ deg. The behaviors of the proposed control schemes are compared with conventional IBVS. Since a very large rotation about the Z-axis (around 140 deg) is considered, the control **A**

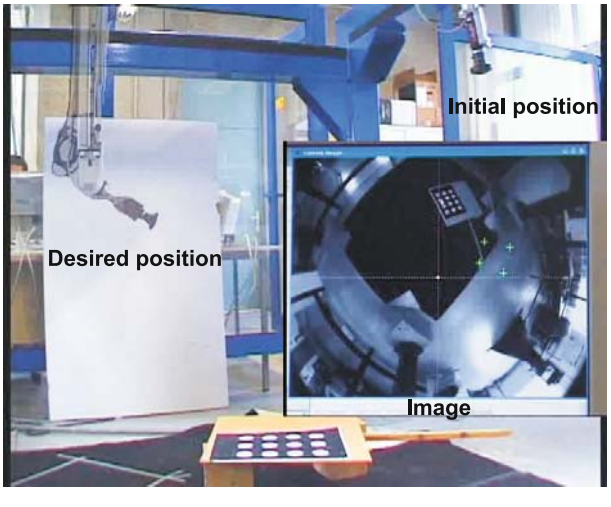


Fig. 16.2 Experimental setup: eye-in-hand configuration

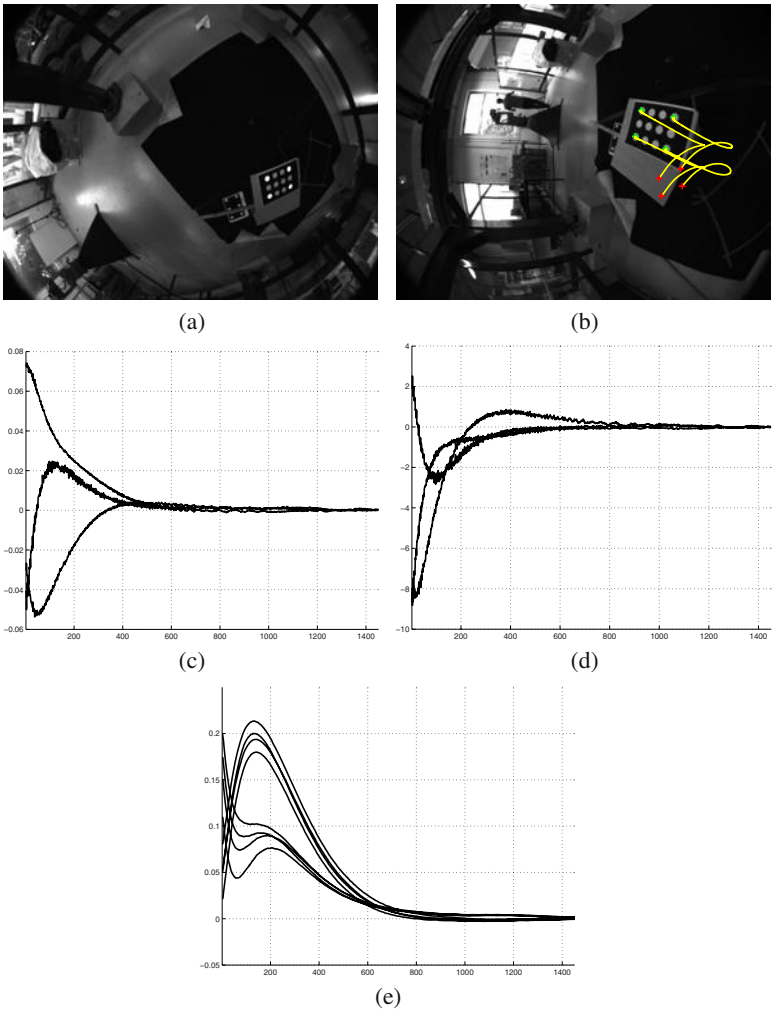


Fig. 16.3 **A**, 2D points-based visual servoing: (a) initial image; (b) desired image and image-points trajectories; (c) translational velocities in m/s; (d) rotational velocities in deg/s; and (e) error vector components

fails, the robot reaching quickly its joint limits. The rotation about the Z-axis is thus reduced to 40 deg for the control **A**.

Figure 16.3 shows the results obtained using the control **A**. The interaction matrix depends on 3D parameters, points coordinates and calibration parameters. If one supposes that the camera-robot system is correctly calibrated and that the measurements are noiseless, the 3D parameters should be accurately estimated to guarantee a quasi-exponential decreasing of the task function **e** (leading to straight line trajectories of the points in the image plane). In this experiment, the 3D parameters

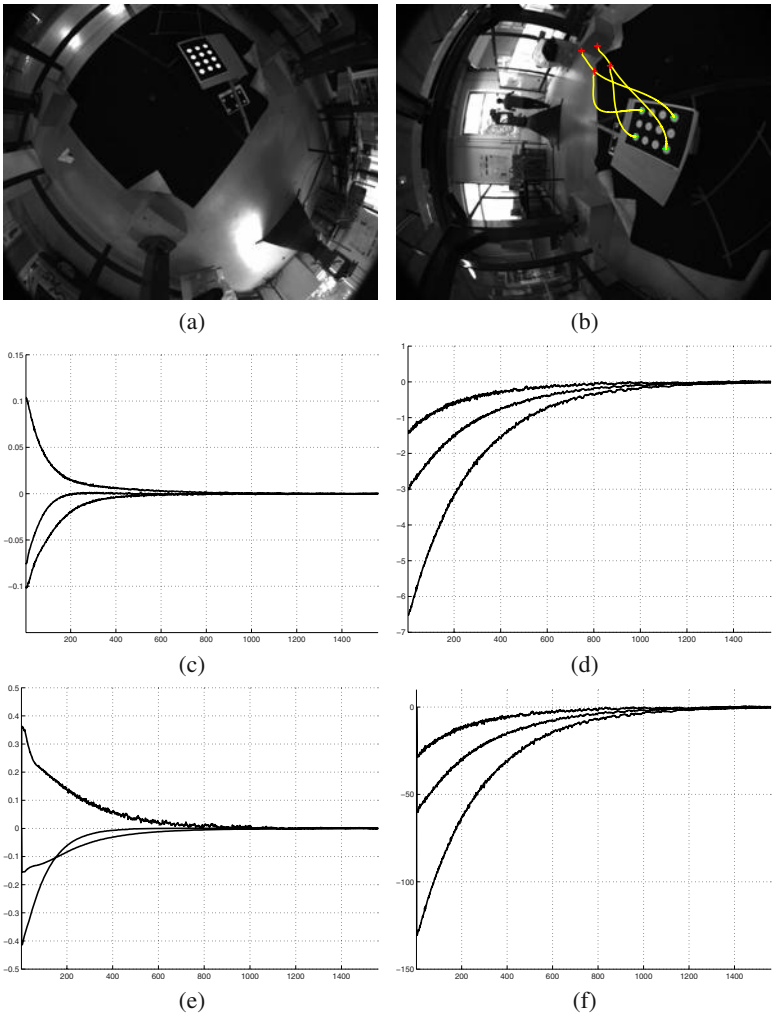


Fig. 16.4 **B**, 2 1/2 D visual servoing: (a) initial image; (b) desired image and image-points trajectories; (c) translational velocities in m/s; (d) rotational velocities in deg/s; and (e) error vector components

ρ_i (which appears in the interaction matrix (16.11)) are set to a constant values $\widehat{\rho}_i = \widehat{\rho^\star}_i$ (where $\widehat{\rho}_i^\star$ denotes the estimated value of ρ_i at the desired configuration). Consequently, the points trajectories are no more straight lines until around the 300^{th} iteration (where ρ_i became very close to $\rho_i^\star$). After the 300^{th} iteration, one can observe that the errors are decreasing exponentially and the image trajectories became roughly straight.

The results obtained with the hybrid control **B**, **C** and **D** are shown in Figures 16.4, 16.5 and 16.6 respectively. The parameter $\rho^\star$ is set to $\widehat{\rho^\star} = 2\rho^\star$ in those cases.

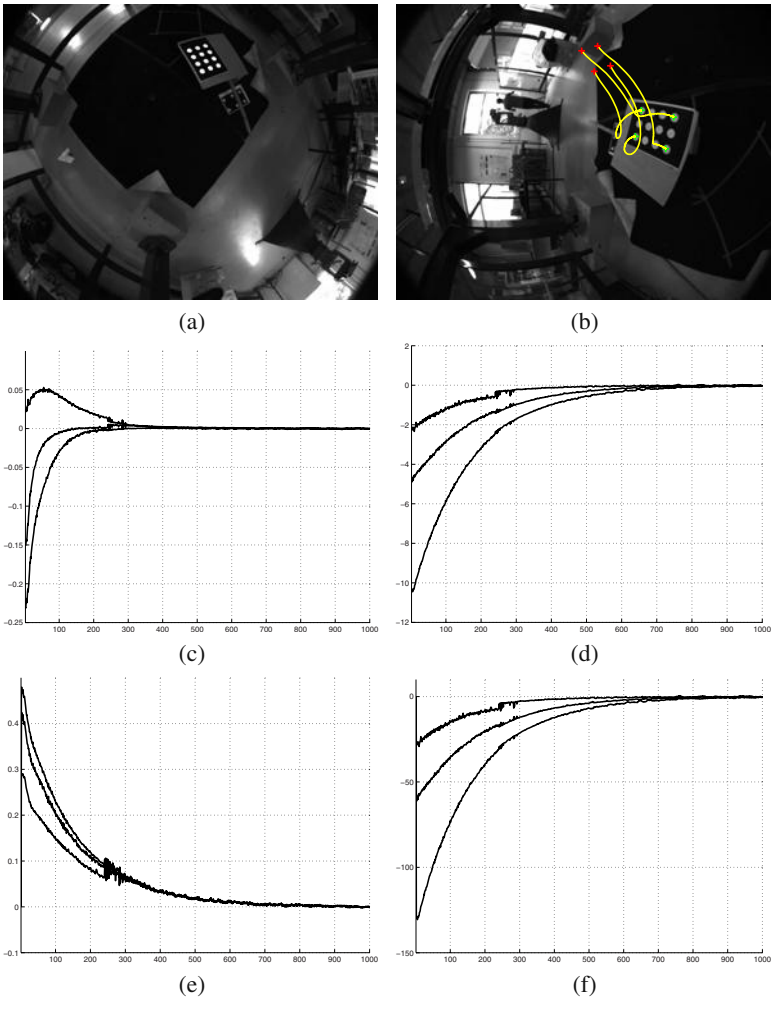


Fig. 16.5 **C**, norm-ratio-based visual servoing: (a) initial image; (b) desired image and image-points trajectories; (c) translational velocities in m/s; (d) rotational velocities in deg/s; and (e) error vector components

It can be first observed that the three control laws allow to achieve the large rotation about the Z-axis (*i.e.* 140 deg) and that, as expected, a rough estimation of the parameter $\rho^\star$ does not affect the system stability. It can be also observed that the decoupling properties have been significantly improved with respect to the 2D points visual servoing. Finally, let us note that, in Figure 16.4(b) the trajectory of the point used to define the 2 1/2 D task function should be a straight line. This is clearly not observed since once again ρ is not correctly estimated.

The control **C** allows to fully decouple translational and rotational motions. However, the computation of the 3D features $\frac{\rho_i}{\rho_i^*}$ increase the sensitivity of the control scheme to measurement noise as it can be observed in Figure 16.5 (see between 200^{th} and 300^{th} iterations). The control law **D** provides very nice decoupling properties (refer to Figure 16.6). In this case, translational velocities are directly related to the visual features (used to control the translational DOF) through a constant diagonal matrix. Furthermore, it can be observed that this control scheme is less sensitive to noise measurement than the previous one.

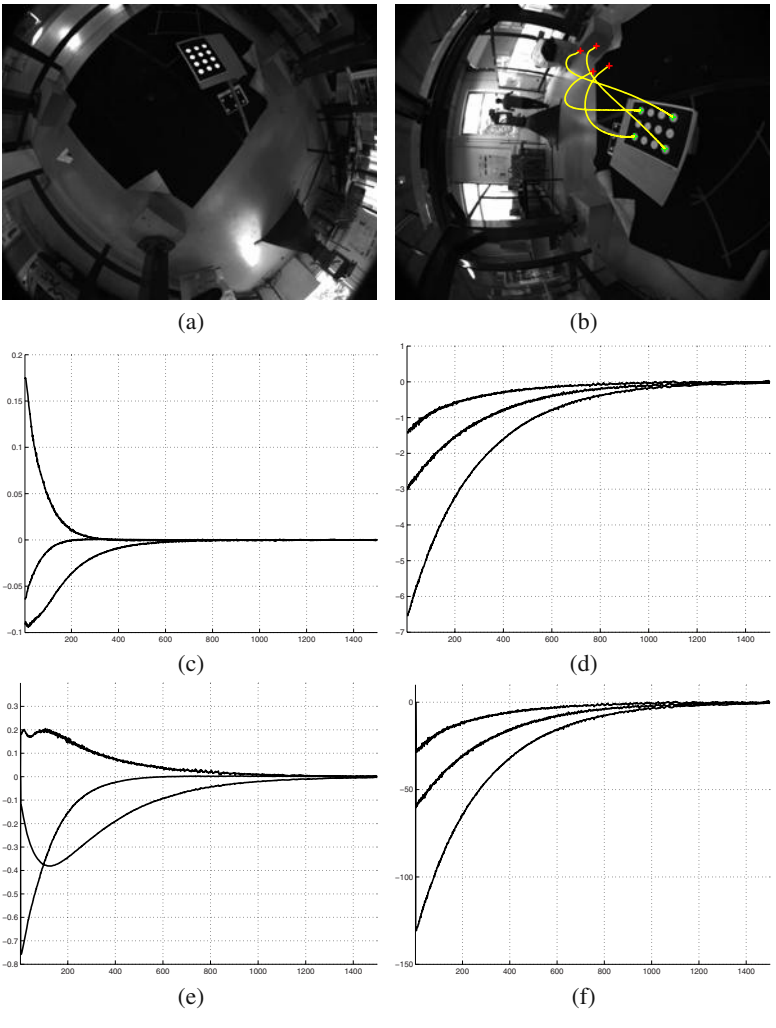


Fig. 16.6 **D**, scaled 3D point-based visual servoing: (a) initial image; (b) desired image and image-points trajectories; (c) translational velocities in m/s; (d) rotational velocities in deg/s; and (e) error vector components

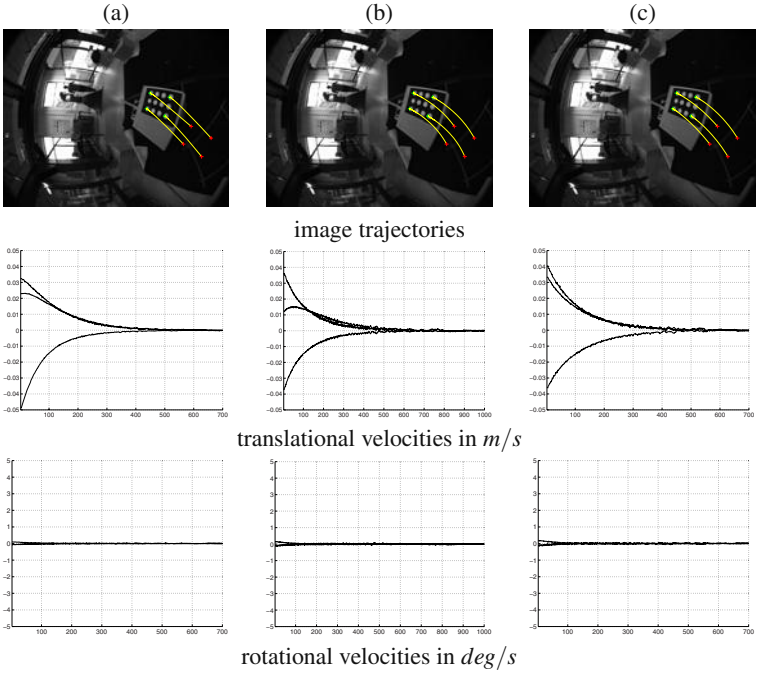


Fig. 16.7 A comparison between the hybrid visual servoing schemes under a pure translation displacement: (a) control scheme in Section 16.3.3.2; (b) control scheme in Section 16.3.3.3; and (c) control scheme in Section 16.3.3.4

Experiment 2. In this set of experiments, the three hybrid schemes are compared when only a translational motion $\mathbf{t} = [80\ 80\ -40]$ cm has to be realized. The results are shown in Figure 16.7. It can be observed that the behavior of the three control schemes is similar. These results confirm also that the control scheme based on the features $\frac{\rho_i}{\rho_i^*}$ seems to be the most sensitive to measurement noises. One can also observe a nonzero rotational velocities at the beginning due to the measurement noises and calibration errors.

Experiment 3. In this set of experiments, only a rotational motion about the Z-axis is considered. The control laws **B**, **C** and **D** are first tested with a huge rotation of 140 deg. In this case, only the control law **C** allows to reach the desired configuration. When using the control laws **B** and **D** the robot reached its joint limits due to the coupling between rotational and translational motions. In the results shown in Figure 16.8, the rotation about the Z-axis is reduced to 90 deg for the control laws **B** and **D**. Finally, the full decoupling between translational and rotational motions provided by the control scheme **C** can be clearly observed.

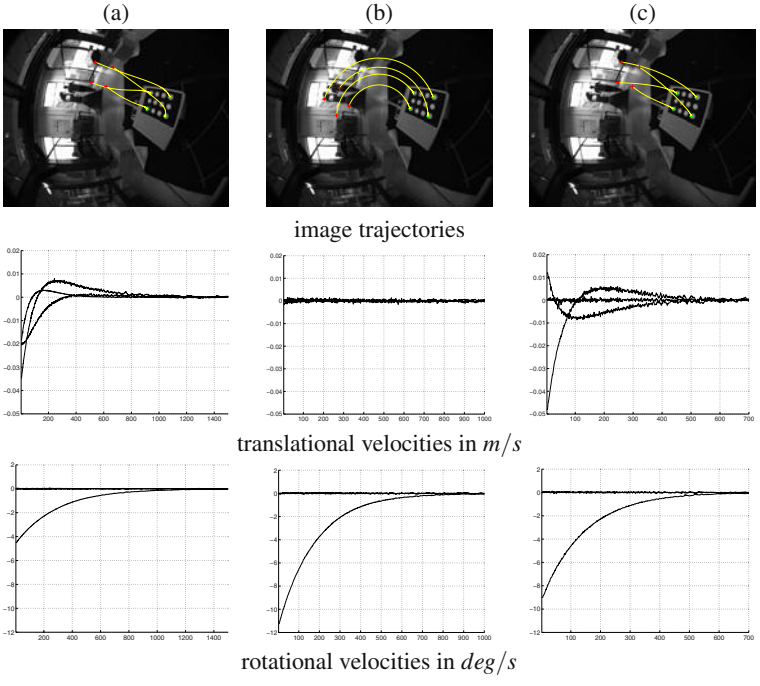


Fig. 16.8 A comparison between the hybrid visual servoing schemes under a pure rotational displacement: (a) control scheme in Section 16.3.3.2; (b) control scheme in Section 16.3.3.3; and (c) control scheme in Section 16.3.3.4

16.5 Conclusion

In this chapter, it has been shown how a generic projection model can be exploited to design vision-based control laws valid for all cameras obeying the unique viewpoint constraint. First, the problem of estimating homographic relationship between two spherical views related to a reference plane has been addressed. Then, three homography-based control schemes have been presented. The task functions are defined to allow as much as possible nice decoupling properties of the control laws. In all cases, the rotational control is achieved using the orientation error extracted from the estimated homography matrix. In the first control scheme, the visual features used to control the translational motions are chosen as the combination of the 2D coordinates of an image point and the ratio of the norms of the corresponding 3D point at the current and desired configurations (which can be computed from the homography matrix). In a second control scheme, a scaled 3D point, computed from the corresponding image point and the homography matrix, is exploited to control efficiently the translations. It allows to obtain properties similar to 3D point-based visual servoing while being model free. The last control law allows to fully decouple translational and rotational motions (the interaction matrix is block-diagonal) by employing three ratios of the norms related to three 3D points. From a practical

point of view, large camera motions can be achieved since the developed control laws are partially or fully decoupled and valid for a large class of wide field of view cameras. Experimental results have confirmed this last point. The stability analysis under modeling errors of the proposed control laws still remain an important theoretical point to be addressed in future works.

References

[1] Barreto, J., Araujo, H.: Geometric properties of central catadioptric line images. In: 7th European Conference on Computer Vision, Copenhagen, Denmark, pp. 237–251 (2002)

[2] Cervera, E., Pobil, A.P.D., Berry, F., Martinet, P.: Improving image-based visual servoing with three-dimensional features. International Journal of Robotics Research 22(10-11), 821–840 (2003)

[3] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman, D., Hager, G., Morse, A. (eds.) The Confluence of Vision and Control. LNCIS, vol. 237, pp. 66–78. Springer, Heidelberg (1998)

[4] Chaumette, F.: Image moments: A general and useful set of features for visual servoing. IEEE Transaction on Robotics and Automation 20(4), 713–723 (2004)

[5] Corke, P.I., Hutchinson, S.A.: A new partitioned approach to image-based visual servo control. IEEE Transaction on Robotics and Automation 17(4), 507–515 (2001)

[6] Courbon, J., Mezouar, Y., Eck, L., Martinet, P.: A generic fisheye camera model for robotic applications. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, California, USA, pp. 1683–1688 (2007)

[7] Espiau, B., Chaumette, F., Rives, P.: A new approach to visual servoing in robotics. IEEE Transaction on Robotics and Automation 8(3), 313–326 (1992)

[8] Faugeras, O., Lustman, F.: Motion and structure from motion in a piecewise planar environment. International Journal of Pattern Recognition and Artificial Intelligence 2(3), 485–508 (1988)

[9] Geyer, C., Daniilidis, K.: A unifying theory for central panoramic systems and practical implications. In: European Conference on Computer Vision, Dublin, Ireland, pp. 159–179 (2000)

[10] Geyer, C., Daniilidis, K.: Mirrors in motion: Epipolar geometry and motion estimation. In: International Conference on Computer Vision, Nice, France, pp. 766–773 (2003)

[11] Hadj-Abdelkader, H., Mezouar, Y., Andreff, N., Martinet, P.: 2 1/2 d visual servoing with central catadioptric cameras. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Canada, pp. 2342–2347 (2005)

[12] Hamel, T., Mahony, R.: Visual servoing of an under-actuated dynamic rigid body system: an image-based approach. IEEE Transaction on Robotics and Automation 18(2), 187–198 (2002)

[13] Iwatsuki, M., Okiyama, N.: A new formulation of visual servoing based on cylindrical coordinates system with shiftable origin. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, pp. 266–273 (2002)

[14] Lee, J.S., Suh, I., You, B.J., Oh, S.R.: A novel visual servoing approach involving disturbance observer. In: IEEE International Conference on Robotics and Automation, Detroit, Michigan, pp. 269–274 (1999)

[15] Malis, E., Chaumette, F.: 2 1/2 d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. International Journal of Computer Vision 37(1), 79–97 (2000)
[16] Malis, E., Chaumette, F.: Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. IEEE Transaction on Robotics and Automation 18(2), 176–186 (2002)
[17] Malis, E., Chaumette, F., Boudet, S.: 2 1/2 d visual servoing. IEEE Transaction on Robotics and Automation 15(2), 238–250 (1999)
[18] Marchand, E., Spindler, F., Chaumette, F.: Visp for visual servoing: a generic software platform with a wide class of robot control skills. IEEE Robotics and Automation Magazine 12(4), 40–52 (2005)
[19] Martinet, P., Daucher, N., Gallice, J., Dhome, M.: Robot control using 3d monocular pose estimation. In: Workshop on New Trends in Image Based Robot Servoing, IEEE/RSJ International Conference on Intelligent Robots and Systems, Grenoble, France, pp. 1–12 (1997)
[20] Mezouar, Y., Chaumette, F.: Path planning for robust image-based control. IEEE Transaction on Robotics and Automation 18(4), 534–549 (2002)
[21] Rives, P., Azinheira, J.: Linear structures following by an airship using vanishing points and horizon line in a visual servoing scheme. In: IEEE International Conference on Robotics and Automation, New Orleans, Louisiana, pp. 255–260 (2004)
[22] Samson, C., Espiau, B., Borgne, M.L.: Robot Control: The Task Function Approach. Oxford University Press, Oxford (1991)
[23] Svoboda, T., Pajdla, T., Hlavac, V.: Motion estimation using central panoramic cameras. In: IEEE Conference on Intelligent Vehicles, Stuttgart, Germany, pp. 335–340 (1998)
[24] Tahri, O., Chaumette, F., Mezouar, Y.: New decoupled visual servoing scheme based on invariants projection onto a sphere. In: IEEE International Conference on Robotics and Automation, Pasadena, CA, pp. 3238–3243 (2008)
[25] Tatsambon Fomena, R., Chaumette, F.: Visual servoing from spheres using a spherical projection model. In: IEEE International Conference on Robotics and Automation, Roma, Italia, pp. 2080–2085 (2007)
[26] Thuilot, B., Martinet, P., Cordesses, L., Gallice, J.: Position based visual servoing: keeping the object in the field of vision. In: IEEE International Conference on Robotics and Automation, Washington DC, USA, pp. 1624–1629 (2002)
[27] Wilson, W., Hulls, C.W., Bell, G.: Relative end-effector control using cartesian position-based visual servoing. IEEE Transaction on Robotics and Automation 12(5), 684–696 (1996)
[28] Zhang, Z., Hanson, A.R.: Scaled euclidean 3d reconstruction based on externally uncalibrated cameras. In: IEEE Symposium on Computer Vision, Coral Gables, FL, pp. 37–42 (1995)

Chapter 17
Sensor-based Trajectory Deformation: Application to Reactive Navigation of Nonholonomic Robots

Florent Lamiraux and Olivier Lefebvre

Abstract. In this chapter, we present a sensor-based trajectory deformation process for nonholonomic robots. The method is based on infinitesimal perturbations of the input functions of the current trajectory. Input perturbation is computed in such a way that a an objective function decreases and that the trajectory initial and final configurations are kept unchanged. The method is then extended to docking for wheeled mobile robots. The final configuration of the deformation process is moved to a configuration in order to make perception fit a docking pattern. The method is demonstrated on mobile robot Hilare 2 towing a trailer.

17.1 Introduction

Navigating multi-body nonholonomic robots in cluttered environments has been a difficult task for a long time, especially when the two following conditions are met:

1. the number of nonholonomic constraints is two or more;
2. the localization uncertainty is less than the clearance to obstacles.

Recent autonomous vehicles competing in the Darpa urban challenge [15] might let the reader think that the problem of autonomous navigation for nonholonomic systems is closed. However, beyond the remarkable work of integration these vehicles are the result of, it should be noticed that these vehicles would not have reached the goal without the recent advances in localization technology. Today, on-the-shelf devices compute in real-time the position of vehicles with an accuracy around the meter. In this context, navigating with a margin of two meters makes the computations relative to motion planning and control much simpler. The challenge is thus more in the field of perception and modelling than in the field of motion planning and control.

Florent Lamiraux and Olivier Lefebvre
LAAS-CNRS, 7 Avenue du Colonel Roche, 31077 Toulouse, France
e-mail: {florent,olefebr}@laas.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 315–334.
springerlink.com

In this chapter, we report on work aiming at addressing the navigation task for systems meeting the two above mentioned conditions. Unlike classical visual servoing methods where the state of the system is the configuration and velocity of the robot, and the input visual perception, the state of our system is a trajectory executable by the robot and the input is a flow of sensor images. This chapter is mostly a compilation of [8] and [14].

To overcome the issue of local minima arising when implementing local control laws, we initially plan an admissible trajectory from the initial configuration of the robot to the goal configuration, using classical approaches of the state of the art in motion planning [1, 11, 19, 3, 20, 12, 9, 13].

Servoing a trajectory instead of a robot state significantly reduces the issue of local minima at the cost of heavy computational load.

Following a planned trajectory can lead to collisions if:

- unexpected obstacles in the environment were not in the map used for planning the motion;
- the map in inaccurate; or
- the localization process is inaccurate.

To overcome these issues, [17] proposed a method that enables a robot to deform on line the path to be followed in order to get away from obstacles detected along the motion. This approach has been extended to the case of a unicycle-like mobile robot in [6] and then to the case of a holonomic mobile manipulator in [2]. In both papers, the geometry of the robot is approximated by a set of balls and no or only one very simple nonholonomic constraint is treated. None of these methods is applicable to more complex nonholonomic systems like car-like robots.

To plan and execute motions in dynamic environments, [5] developed the concept of velocity obstacles, defining the set of forbidden velocities given the velocity of the obstacles. This concept is used in [10] to perform local goal oriented obstacle avoidance. This technique is particularly efficient in environments where a lot of obstacles are moving since the velocity of the obstacles is taken into account in the avoidance strategy. However, it is based on very simple models of robot and obstacles: they all are spherical. This simplification forbids applications for multibody mobile robots moving in very cluttered environments where the robot needs to pass very close to the obstacles.

In this chapter, we describe a generic approach of trajectory deformation applicable to any nonholonomic system. We assume that a first collision-free trajectory has been computed for the robot in the global frame. When the robot follows the trajectory, on-board sensors, for instance laser scanners, detect surrounding obstacles and map them in the global frame. If an obstacle not present in the map is detected, it can be in collision with the initial trajectory. If the localization of the robot is inaccurate, or if the map is inexact, obstacles of the map might be seen in collision with the initial trajectory by the sensors. The method we describe in this chapter enables the robot to deform the initial trajectory in order to move it away from obstacles and make the current trajectory collision-free . The current trajectory thus changes along time. As a trajectory is a mapping from an interval of real numbers into the

configuration space of the robot, we naturally model a trajectory deformation process as a mapping of two real variables s and τ into the configuration space. τ can be considered as time (or more generally as an increasing function of time), while s is the abscissa along each trajectory.

The chapter is organized as follows. Section 17.2 defines trajectory deformation as an infinite-dimensional dynamic control system the state of which is a trajectory. In Section 17.3, we describe an iterative algorithm controlling the deformation process to make an optimization criterion decrease. In Section 17.4, the trajectory deformation algorithm is applied to mobile robot Hilare 2 towing a trailer. In Section 17.5, the method is extended to perform docking for nonholonomic robots.

17.2 Nonholonomic Trajectory Deformation as a Dynamic Control System

A trajectory for a robotic system is usually represented by a mapping from an interval of $\mathbb{R}$ into the configuration space of the system. In this section, we introduce the notion of trajectory deformation as a mapping from an interval of $\mathbb{R}$ into the set of trajectories. Equivalently, a trajectory deformation is a mapping from two intervals into the configuration space as explained later in this section.

17.2.1 Admissible Trajectories

A nonholonomic system of dimension n is characterized by a set of $k < n$ vector fields $\mathbf{X}_1(\mathbf{q}),...,\mathbf{X}_k(\mathbf{q})$, where $\mathbf{q} \in \mathscr{C} = \mathbb{R}^n$ is the configuration of the system. For each configuration $\mathbf{q}$, the set admissible velocities of the system is the set of linear combinations of the $\mathbf{X}_i(\mathbf{q})$. A trajectory $\mathbf{q}(s)$ is a smooth curve in the configuration space defined over an interval $[0,S]$. A trajectory is said to be admissible if and only if there exists a k-dimensional smooth vector valued mapping $\mathbf{u} = (u_1,...,u_k)$ defined over $[0,S]$ and such that:

$$\forall \in [0,S] \quad \mathbf{q}'(s) = \sum_{i=1}^{k} u_i(s)\mathbf{X}_i(\mathbf{q}(s)) \tag{17.1}$$

where from now on, $'$ denotes the derivative with respect to s.

17.2.2 Admissible Trajectory Deformation

We call trajectory deformation a mapping from a subset $[0,S] \times [0,\infty)$ of $\mathbb{R}^2$ to the configuration space of the system:

$$(s,\tau) \rightarrow \mathbf{q}(s,\tau).$$

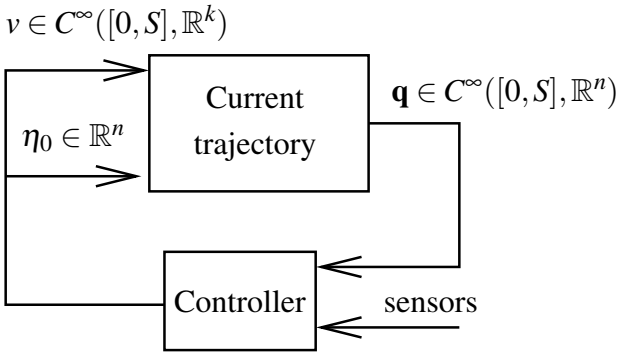


Fig. 17.1 A trajectory deformation process can be modelled as a dynamic control system of time τ. At each time, the state is a feasible trajectory $\mathbf{q}$, the input is a pair $(\eta_0, \mathbf{v})$ that uniquely defines the time derivative of the state. The trajectory deformation algorithm we describe in this chapter can be considered as a closed-loop controller that computes the input of the dynamic control system with respect to the current trajectory and a task to achieve, for instance avoiding obstacles, based on perceptual data

For each value of τ, $s \to \mathbf{q}(s,\tau)$ is a trajectory. $s \to \mathbf{q}(s,0)$ is called the *initial trajectory*. In order to keep notation light and intuitive, we use the same notation $\mathbf{q}$ to denote configurations, trajectories and trajectory deformations. We are interested in deformations $\mathbf{q}(s,\tau)$ composed of only admissible trajectories. Such deformations satisfy the following constraint: there exists a k-dimensional vector valued smooth mapping $\mathbf{u} = (u_1,...,u_k)$ defined over $[0,S] \times [0,\infty)$ such that $\forall (s,\tau) \in [0,S] \times [0,\infty)$

$$\frac{\partial \mathbf{q}}{\partial s}(s,\tau) = \sum_{i=1}^{k} u_i(s,\tau)\mathbf{X}_i(\mathbf{q}(s,\tau)). \tag{17.2}$$

For each value of τ, $s \to \mathbf{u}(s,\tau)$ is the input function of trajectory $s \to \mathbf{q}(s,\tau)$. The above equation simply expresses constraint (17.1) for each trajectory of the deformation. As well as a trajectory is uniquely defined by the initial configuration and the input function, a trajectory deformation is uniquely defined by the initial configuration $\mathbf{q}(0,\tau)$ of each trajectory and by input functions $u_i(s,\tau)$.

By differentiating (17.2), we get a relation between the input variation $\frac{\partial \mathbf{u}}{\partial \tau}$ and the infinitesimal trajectory deformation when the deformation parameter τ increases:

$$\frac{\partial^2 \mathbf{q}}{\partial s \partial \tau}(s,\tau) = \sum_{i=1}^{k} \left(\frac{\partial u_i}{\partial \tau}(s,\tau)\mathbf{X}_i(\mathbf{q}(s,\tau)) + u_i(s,\tau)\frac{\partial \mathbf{X}_i}{\partial \mathbf{q}}(\mathbf{q}(s,\tau))\frac{\partial \mathbf{q}}{\partial \tau}(s,\tau) \right)$$

We call respectively *input perturbations* and *direction of deformation* the following vector valued functions:

$$\mathbf{v}(s,\tau) \triangleq \frac{\partial \mathbf{u}}{\partial \tau}(s,\tau)$$
$$\eta(s,\tau) \triangleq \frac{\partial \mathbf{q}}{\partial \tau}(s,\tau).$$

With this notation, the above equation becomes

$$\eta'(s,\tau) = A(s,\tau)\eta(s,\tau) + B(s,\tau)\mathbf{v}(s,\tau) \tag{17.3}$$

where $A(s,\tau)$ is the following $n \times n$ matrix:

$$A(s,\tau) = \sum_{i=1}^{k} u_i(s,\tau)\frac{\partial X_i}{\partial \mathbf{q}}(\mathbf{q}(s,\tau))$$

and $B(s,\tau)$ is the $n \times k$ matrix the columns of which are the control vector fields:

$$B(s,\tau) = \left(X_1(\mathbf{q}(s,\tau)) \cdots X_k(\mathbf{q}(s,\tau)) \right).$$

According to (17.3), the derivative with respect to τ of the trajectory of parameter τ is related to the input perturbation through a linear dynamic system. This system is in fact the linearized system of (17.1) about the trajectory of parameter τ: $s \rightarrow \mathbf{q}(s,\tau)$. For a given trajectory $\mathbf{q}(s,\tau)$ of input $\mathbf{u}(s,\tau)$ and for any input perturbation $\mathbf{v}(s,\tau)$, and any initial condition $\eta_0 = \eta(0,\tau)$ we can integrate (17.3) with respect to s to get the corresponding direction of deformation $\eta(s,\tau)$.

A trajectory deformation process for nonholonomic systems can thus be considered as a dynamic control system where:

- τ is the time;
- $s \rightarrow \mathbf{q}(s,\tau)$ is the state;
- $(\eta_0, s \rightarrow \mathbf{v}(s,\tau)$ is the input.

17.2.3 Potential Field and Inner Product

The trajectory deformation method produces at each time τ a vector η_0 and a function $s \rightarrow \mathbf{v}(s,\tau)$ over $[0,S]$ in such a way that the deformation process achieves a specified goal. This goal is expressed in terms of a scalar value to minimize over the set of feasible trajectories. The scalar value associated to a trajectory is defined by integration of a potential field U over the configuration space. We denote by $V(\tau)$ the potential value of trajectory $s \rightarrow \mathbf{q}(s,\tau)$:

$$V(\tau) \triangleq \int_0^S U(\mathbf{q}(s,\tau))ds.$$

If the goal to achieve is to avoid obstacles, as in [18, 7, 1], the configuration space potential field is defined in such a way that the value is high for configurations close to obstacles and low for configuration far from obstacles. Thus trajectories

going close to obstacles have a high scalar value and trajectories staying far from obstacles have a low scalar value.

The variation of the trajectory scalar value with respect to τ is related to $\eta(s,\tau)$ by the following expression:

$$\frac{dV}{d\tau}(\tau) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s,\tau))^T \eta(s,\tau) ds.$$

The principle of the trajectory deformation method consists in choosing $(\eta_0, \mathbf{v}(s,\tau))$ in such a way that $\frac{dV}{d\tau}(\tau)$ is negative. Let us notice that the space of vector-valued functions defined over interval $[0,S]$ is a Hilbert space, the inner product of which is defined by

$$(f|g)_{L^2} \triangleq \int_0^S f(s)^T g(s) ds. \tag{17.4}$$

With this definition, the variation of the trajectory scalar value along a direction of deformation can be rewritten as

$$\frac{dV}{d\tau}(\tau) = \left(\frac{\partial U}{\partial \mathbf{q}} \circ \mathbf{q} | \eta\right)$$

where $\circ$ denotes the composition operator. Let us notice that integration is performed over variable s only. According to this expression, $\eta = -(\frac{\partial U}{\partial \mathbf{q}} \circ \mathbf{q})$ is at equivalent L^2-norm the direction of deformation that minimizes $\frac{dV}{d\tau}$. Unfortunately, this value of η is not an admissible direction of deformation (*i.e.* a solution of system (17.3)). A solution could be obtained by orthogonally projecting $-(\frac{\partial U}{\partial \mathbf{q}} \circ \mathbf{q})$ over the linear subspace of admissible directions of deformation. However, the projection of a vector over an infinite-dimensional subspace does not necessarily exist.

To overcome this problem, we will restrict the input perturbation to a finite-dimensional subspace in the following section.

17.3 Nonholonomic Trajectory Deformation Algorithm

Based on the theoretical framework established in the previous section, we build in this section the trajectory deformation algorithm for nonholonomic systems. Starting from an initial admissible trajectory $\mathbf{q}(s,0)$, the algorithm iteratively computes a sequence of admissible trajectories $s \to \mathbf{q}(s,\tau_j)$ for discretized values τ_j of τ where j is an integer. At each iteration of the algorithm, a direction of deformation $\eta(s,\tau_j)$ is generated based on the configuration space potential field U and a new trajectory $\mathbf{q}(s,\tau_{j+1})$ is computed as

$$\mathbf{q}(s,\tau_{j+1}) = \mathbf{q}(s,\tau_j) + \Delta\tau_j \, \eta(s,\tau_j) \tag{17.5}$$

$$\tau_{j+1} = \tau_j + \Delta\tau_j \tag{17.6}$$

where $\Delta\tau_j$ is the discretization step. Let us notice that the above formula is a first-order approximation in τ. In the rest of this section, we describe the different steps of

the algorithm. In Section 17.3.1, we compute $\eta(s,\tau_j)$ by restricting input perturbation to a finite-dimensional subspace of functions. This restriction enables us in Section 17.3.2 to take into account boundary conditions that force the initial and final configuration of the deformation interval to remain unchanged. In Section 17.3.3, we explain how to compute the direction of deformation that minimizes the variation of the trajectory scalar value under constant L^2-norm. The first order approximation 17.5 induces deviations of the nonholonomic constraints. Section 17.3.4 addresses this issue and proposes a correction of this deviation.

17.3.1 Finite-dimensional Subspace of Input Perturbations

As explained in Section 17.2, the control variables of a trajectory deformation process are the input perturbation $\mathbf{v}$ and the initial condition η_0. $s \rightarrow \mathbf{v}(s,\tau_j)$ belongs to the infinite-dimensional space of smooth vector-valued functions defined over $[0,S]$. To simplify the control of the trajectory deformation, we choose to restrict $\mathbf{v}$ to a finite-dimensional subspace of functions. This restriction will make the boundary conditions introduced later in Section 17.3.2 easier to deal with. Let p be a positive integer. We define $\mathbf{e}_1,...,\mathbf{e}_p$, a set of smooth linearly independent vector-valued functions of dimension k, defined over $[0,S]$:

$$\mathbf{e}_i : [0,S] \rightarrow \mathbb{R}^k.$$

Various choices are possible for the $\mathbf{e}_i$'s (*e.g.* truncated Fourier series, polynomials, *etc*) [16, 4, 3]. For each of these functions, we define $\mathbf{E}_i(s,\tau_j)$ as the solution of system (17.3) with initial condition $\eta_0 = 0$ and with $\mathbf{e}_i(s)$ as input:

$$\mathbf{E}_i'(s,\tau_j) = A(s,\tau_j)\mathbf{E}_i(s,\tau_j) + B(s,\tau_j)\mathbf{e}_i(s) \tag{17.7}$$

$$\mathbf{E}_i(0,\tau_j) = 0 \tag{17.8}$$

where matrices A and B are defined in Section 17.2.2. Let us notice that unlike $\mathbf{e}_i$, $\mathbf{E}_i$ depends on τ_j since system (17.3) depends on the current trajectory.

If we restrict $\mathbf{v}(s,\tau_j)$ in the set of functions spanned by the $\mathbf{e}_i$'s, that is for any vector $\lambda = (\lambda_1,...,\lambda_p)$:

$$\mathbf{v}(s,\tau_j) = \sum_{i=1}^{p} \lambda_i e_i(s) \tag{17.9}$$

as (17.3) is linear, the direction of deformation η corresponding to $\mathbf{v}$ is the same linear combination of solutions $\mathbf{E}_i$

$$\eta(s,\tau_j) = \sum_{i=1}^{p} \lambda_i \mathbf{E}_i(s,\tau_j). \tag{17.10}$$

Using this restriction, the input perturbation $\mathbf{v}$ is uniquely defined by vector λ.

17.3.2 Boundary Conditions

We wish the deformation process not to modify the initial and goal configurations of the trajectory. We thus impose the following boundary conditions:

$$\begin{aligned} \forall j > 0, \quad \mathbf{q}(0,\tau_j) &= \mathbf{q}(0,0) \\ \mathbf{q}(S,\tau_j) &= \mathbf{q}(S,0). \end{aligned}$$

These constraints are equivalent to

$$\forall j > 0, \quad \eta(0,\tau_j) = 0 \tag{17.11}$$

$$\eta(S,\tau_j) = 0. \tag{17.12}$$

(17.8) and (17.10) ensure us that the first constraint (17.11) is satisfied. The second constraint (17.12) together with (17.10) becomes a linear constraint over vector λ:

$$L\lambda = 0 \tag{17.13}$$

where L is a $n \times p$-matrix the columns of which are the $\mathbf{E}_i(S,\tau_j)$'s:

$$L = \left(\mathbf{E}_1(S,\tau_j) \cdots \mathbf{E}_p(S,\tau_j). \right)$$

Let us notice that in general, the dimension of the subspace of solutions of the above linear system is equal to $p-n$ and therefore p must be bigger than n. The problem is now to choose a vector λ satisfying the above linear constraint and generating a direction of deformation that makes the current trajectory move away from obstacles. We address this issue in the following section.

17.3.3 Direction of Deformation That Makes Trajectory Scalar Value Decrease

As explained in Section 17.2.3, a potential field U is defined over the configuration space. This potential field defines by integration a scalar valued function V over the space of trajectories.

Given a vector $\lambda \in \mathbb{R}^p$, the variation of the trajectory scalar value induced by direction of deformation η defined by (17.10) is given by

$$\frac{dV}{d\tau}(\tau_j) = \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s,\tau_j))^T \eta(s,\tau_j) ds \tag{17.14}$$

$$= \sum_{i=1}^{p} \lambda_i \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s,\tau_j))^T \mathbf{E}_i(s,\tau_j) ds. \tag{17.15}$$

Let us define the coefficients

$$\mu_i \triangleq \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s,\tau_j))^T \mathbf{E}_i(s,\tau_j) ds.$$

These coefficients represent the variation of the trajectory scalar value induced by each direction of deformation $\mathbf{E}_i$. With these coefficients, (17.15) can be rewritten as

$$\frac{dV}{d\tau}(\tau_j) = \sum_{i=1}^{p} \lambda_i \mu_i. \tag{17.16}$$

Thus, if we choose

$$\lambda_i = -\mu_i \tag{17.17}$$

we get a trajectory deformation $\eta(.,\tau_j)$ that keeps the kinematic constraints satisfied and that makes the trajectory scalar value decrease. Indeed:

$$\frac{dV}{d\tau}(\tau_j) = -\sum_{i=1}^{p} \mu_i^2 \leq 0.$$

We denote by λ^0 this value of vector λ. Of course, nothing ensures us that λ^0 satisfies the boundary conditions (17.13).

17.3.3.1 Projection over the Subspace of Boundary Conditions

(17.13) states that the set of vectors λ satisfying the boundary conditions is a linear subspace of $\mathbb{R}^p$. To get such a vector that we denote by $\bar{\lambda}$, we project λ^0 over this subspace:

$$\bar{\lambda} = (I_p - L^+ L)\lambda^0$$

where I_p is the identity matrix of size p and L^+ is the Moore–Penrose pseudo-inverse of matrix L.

It can be easily verified that $\bar{\lambda}$ satisfies the following properties:

1. $L\bar{\lambda} = 0$;
2. $\eta = \sum_{i=1}^{p} \bar{\lambda}_i \mathbf{E}_i$ makes the trajectory scalar value decrease.

17.3.3.2 A Better Direction of Deformation

Let us recall that (17.5) is an approximation of order 1 with respect to τ. For this reason, $\Delta\tau_j \|\eta\|_\infty$ with $\|\eta\|_\infty \triangleq \max_{s\in[0,S]} \|\eta(s,\tau_j)\|$ needs to be small. $\Delta\tau_j$ is thus chosen in such a way that $\Delta_{\tau_j}\|\eta\|_\infty$ is upper bounded by a positive given value η_{max}. The way the λ_i's are chosen in (17.17) is not optimal in this respect. Indeed, the goal we aim at at each iteration is to make the trajectory scalar value V decrease at most for constant $\|\eta\|_\infty$. Therefore the optimal value of λ realizes the following minimum:

$$\min_{\|\eta\|_\infty=1} \frac{dV}{d\tau}(\tau_j) = \min_{\|\sum_{i=1}^{p} \lambda_i \mathbf{E}_i\|_\infty=1} \sum_{i=1}^{p} \mu_i \lambda_i$$

Unfortunately, this value of vector λ is very difficult to determine since $\|.\|_\infty$ is not a Euclidean norm. Instead, we compute

$$\min_{\|\sum_{i=1}^{p} \lambda_i \mathbf{E}_i\|_{L^2}=1} \sum_{i=1}^{p} \mu_i \lambda_i.$$

This is a better approximation than (17.17).

The idea of the computation is to express η in an L^2-orthonormal basis in such a way that the above sum becomes the inner product between two vectors. Let us build from $(\mathbf{E}_1, \cdots, \mathbf{E}_p)$ an orthonormal basis $(\mathbf{F}_1, \cdots, \mathbf{F}_p)$ using Gram–Schmidt orthonormalization procedure. Let P be the corresponding $p \times p$ matrix of change of coordinates (the j-th column of P is the vector of coordinates of $\mathbf{F}_j$ expressed in $(\mathbf{E}_1, \cdots, \mathbf{E}_p)$). If we express η in $(\mathbf{F}_1, \cdots, \mathbf{F}_p)$ instead of $(\mathbf{E}_1, \cdots, \mathbf{E}_p)$, (17.10) becomes

$$\eta(s, \tau_j) = \sum_{i=1}^{p} \lambda_i^{\perp} \mathbf{F}(s, \tau_j)$$

and (17.16) becomes

$$\frac{dV}{d\tau}(\tau_j) = \sum_{i=1}^{p} \lambda_i^{\perp} \mu_i^{\perp} = \left(\mu^{\perp} | \eta\right)_{L^2} \tag{17.18}$$

with

$$\mu_i^{\perp} \triangleq \int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s, \tau_j))^T \mathbf{F}_i(s, \tau_j) ds$$

and $\mu^{\perp} = \sum_{i=1}^{p} \mu_i^{\perp} \mathbf{F}_i$. The second equality in (17.18) holds since $(\mathbf{F}_1, \cdots, \mathbf{F}_p)$ is L_2-orthonormal. At equivalent L_2-norm, $\eta = -\mu^{\perp}$ (*i.e.* $\lambda_i = -\mu_i^{\perp}$) is the direction of deformation that minimizes $\frac{dV}{d\tau}(\tau_j)$. In fact we do not evaluate functions $\mathbf{F}_l$'s, but only matrix P. The expression of η in basis $(\mathbf{E}_1, \cdots, \mathbf{E}_p)$ is given by vector

$$\lambda = P\lambda^{\perp} = PP^T \lambda^0. \tag{17.19}$$

Using expression of η in the orthonormal basis $(\mathbf{F}_1, \cdots, \mathbf{F}_p)$, the expression in $(\mathbf{E}_1, \cdots, \mathbf{E}_p)$ of the orthogonal projection of the above η over the subspace of vectors satisfying the boundary conditions (17.12) becomes

$$\bar{\lambda} = (I_p - P(LP)^{+} L) PP^T \lambda^0$$

Using the above optimal direction of deformation makes the trajectory deformation algorithm behave much better. It can be explained by the fact that this choice makes the trajectory scalar value decrease faster and thus is more efficient to get away from obstacles.

17.3.4 Nonholonomic Constraint Deviation

Approximation (17.5) induces a side effect: after a few iterations, the nonholonomic constraints are not satisfied anymore and the trajectory becomes non admissible.

We call this effect the nonholonomic constraint deviation. The goal of this section is to correct this deviation. If a trajectory is not admissible, the velocity along this trajectory is not contained in the linear subspace spanned by the k control vector fields and condition (17.1) does not hold.

17.3.4.1 Extended Dynamic System

To take into account this issue, for each configuration $\mathbf{q}$, we add $n-k$ vector fields $\mathbf{X}_{k+1}(\mathbf{q}),\cdots,\mathbf{X}_n(\mathbf{q})$ to the k control vector fields of the system in such a way that $\mathbf{X}_1(\mathbf{q}),\cdots,\mathbf{X}_n(\mathbf{q})$ span $\mathbb{R}^n$. We define the extended system as the system controlled by all these vector fields:

$$\mathbf{q}' = \sum_{i=1}^{n} u_i \mathbf{X}_i(\mathbf{q}). \tag{17.20}$$

System (17.20) is not subject to any kinematic constraint. A trajectory $\mathbf{q}(s)$ of system (17.20) is admissible for system (17.1) if and only if for any $j \in \{k+1,\cdots,n\}$ and any $s \in [0,S], u_j(s) = 0$.

In Section 17.2, we deformed a given trajectory, admissible for (17.1) by perturbing the input functions $u_1(s,\tau),\cdots,u_k(s,\tau)$ of this trajectory in order to avoid obstacles. In this section, we consider an initial trajectory not necessarily admissible and we compute input perturbations that make $u_{k+1}(s,\tau),\cdots,u_n(s,\tau)$ uniformly tend toward 0 as τ grows.

From now on, we denote by $\bar{\mathbf{u}}(s,\tau) = (u_1(s,\tau),\cdots,u_n(s,\tau))$ the input function of system (17.20) and by $\bar{\mathbf{v}}(s,\tau) = (v_1(s,\tau),\cdots,v_n(s,\tau))$ the perturbation of these input functions:

$$\forall i \in \{1,\cdots,n\},\ \ v_i(s,\tau) = \frac{\partial u_i}{\partial \tau}(s,\tau).$$

The relation between the input perturbation $\bar{\mathbf{v}}$ and the direction of deformation η is similar as in Section 17.2:

$$\eta'(s,\tau) = \bar{A}(s,\tau)\eta(s,\tau) + \bar{B}(s,\tau)\bar{\mathbf{v}}(s,\tau) \tag{17.21}$$

but now, $\bar{A}(s,\tau)$ and $\bar{B}(s,\tau)$ are both $n \times n$ matrices:

$$\bar{A} = \sum_{i=1}^{n} u_i \frac{\partial \mathbf{X}_i}{\partial \mathbf{q}}(\mathbf{q}) \ \text{ and } \ \bar{B} = (BB^{\perp}) \tag{17.22}$$

where $B^{\perp} = (\mathbf{X}_{k+1}(\mathbf{q})\cdots\mathbf{X}_n(\mathbf{q}))$ is the matrix the column of which are the additional vector fields. With this notation, (17.21) can be rewritten as

$$\eta'(s,\tau) = \bar{A}(s,\tau)\eta(s,\tau) + B(s,\tau)\mathbf{v}(s,\tau) + B^{\perp}(s,\tau)\mathbf{v}^{\perp}(s,\tau) \tag{17.23}$$

where $\mathbf{v}^{\perp}(s,\tau) = (v_{k+1}(s,\tau),\cdots,v_n(s,\tau))$.

17.3.4.2 Correction of Nonholonomic Deviation

In order to make $u_{i+1}(s,\tau),\cdots,u_n(s,\tau)$ tend toward 0 as τ increases, we apply the following linear control:

$$\forall i \in \{k+1,\cdots,n\}, \forall s \in [0,S],\ \ v_i(s,\tau) = -\alpha u_i(s,\tau)$$

where α is a positive constraint. We denote by η_1 the corresponding direction of deformation for $\tau = \tau_j$:

$$\eta_1'(s,\tau_j) = \bar{A}(s,\tau_j)\eta_1(s,\tau_j) + B^{\perp}(s,\tau_j)\mathbf{v}^{\perp}(s,\tau_j) \tag{17.24}$$

$$\eta_1(0,\tau_j) = 0. \tag{17.25}$$

17.3.4.3 Deformation due to Obstacles

Following the procedure described in sections 17.3.1 and 17.3.3, we restrict input functions $(\mathbf{v}_1,\cdots,\mathbf{v}_k)$ to the finite dimensional subspace of functions spanned by $(\mathbf{e}_1,\cdots,\mathbf{e}_p)$ and we compute $\lambda = (\lambda_1,\cdots,\lambda_p)$ according to (17.19). We denote by η_2 the direction of deformation obtained with these coefficients:

$$\eta_2(s,\tau_j) = \sum_{i=1}^{p} \lambda_i \mathbf{E}_i(s,\tau_j)$$

where the $\mathbf{E}_i$ are now solution of system:

$$\mathbf{E}_i'(s,\tau_j) = \bar{A}(s,\tau_j)\mathbf{E}_i(s,\tau_j) + B(s,\tau_j)\mathbf{e}(s) \tag{17.26}$$

$$\mathbf{E}_i(0,\tau_j) = 0. \tag{17.27}$$

17.3.4.4 Boundary Conditions

We wish the sum of η_1 and η_2 satisfies boundary conditions (17.13) and (17.14). Again, (17.13) is trivially satisfied. (17.14) is an affine constraint over vector λ:

$$\eta_2(S,\tau_j) = L\lambda = -\eta_1(S,\tau_j) \tag{17.28}$$

where L is the matrix defined in Section 17.3.2. Following the same idea as in Section 17.3.2, we project vector λ over the affine subspace satisfying (17.28):

$$\bar{\lambda} = -P(LP)^{+}\eta_1(S,\tau_j) + (I_p - P(LP)^{+}L)\lambda\,.$$

We then get a direction of deformation satisfying the boundary conditions and making the component of the velocity along additional vector fields converge toward 0:

$$\eta(s,\tau_j) = \sum_{i=1}^{p} \bar{\lambda}_i \mathbf{E}_i(s,\tau_j) + \eta_1(s,\tau_j).$$

Table 17.1 Trajectory deformation algorithm: at each step, the direction of deformation $\eta(s,\tau_j)$ is computed given the current trajectory $\mathbf{q}(s,\tau_j)$ and the potential field defined by obstacles

Algorithm : Trajectory deformation for nonholonomic systems

/* current trajectory = initial trajectory */
$j=0$; $\tau_j=0$ **while** $\mathbf{q}(s,\tau_j)$ in collision {

compute $\bar{A}(s,\tau_j)$ and $\bar{B}(s,\tau_j)$ for $s\in[0,S]$
/* correction of nonholonomic deviation */
for k in $\{k+1,\ldots,n\}$ {
compute $u_i(s,\tau_j)$
compute $v_i(s,\tau_j)=-\alpha u_i(s,\tau_j)$
compute $\eta_1(s,\tau_j)$ using (17.24)
}

/* potential gradient in configuration space */
for i in $\{1,\ldots,p\}$ {
compute $\mathbf{E}_i(s,\tau_j)$ by integrating (17.26)
}
compute $\frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s,\tau_j))$ for $s\in[0,S]$
for i in $\{1,\ldots,p\}$ {
compute $\lambda_i^0=-\int_0^S \frac{\partial U}{\partial \mathbf{q}}(\mathbf{q}(s,\tau_j))^T\mathbf{E}_i(s,\tau_j)ds$
}

/* orthonormalization*/
compute matrix P using Gram-Schmidt procedure

/* projection of λ over boundary conditions */
compute $\bar{\lambda}=-P(LP)^+\eta_1(S,\tau_j)+(I_p-P(LP)^+L)\lambda$

/* compute and apply deformation */
compute $\eta(s,\tau_j)=\sum_{i=1}^p \bar{\lambda}_i\mathbf{E}_i(s,\tau_j)$ for $s\in[0,S]$
$\mathbf{q}(s,\tau_j)\leftarrow\mathbf{q}(s,\tau_j)+\Delta\tau\,\eta(s,\tau_j)$ for $s\in[0,S]$
}

Table 17.1 summarizes an iteration of the trajectory deformation algorithm for nonholonomic systems.

17.4 Application to Mobile Robot Hilare 2 Towing a Trailer

In this section, we briefly illustrate the developments of the previous section by applying them to mobile robot Hilare 2 towing a trailer (see Figure 17.2). We refer the reader to [8] for more details.

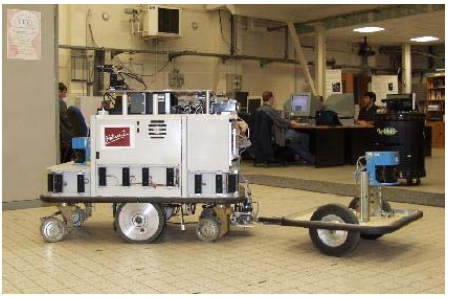


Fig. 17.2 Mobile robot Hilare 2 towing a trailer

A configuration of this robot is represented by $\mathbf{q} = (x, y, \theta, \varphi)$ where (x, y) is the position of the center of the robot, θ is the orientation of the robot and φ is the orientation of the trailer with respect to the robot. The control vector fields are

$$\mathbf{X}_1 = \begin{pmatrix} \cos\theta \\ \sin\theta \\ 0 \\ -\frac{1}{l_t}\sin\varphi \end{pmatrix} \quad \mathbf{X}_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 - \frac{l_r}{l_t}\cos\varphi \end{pmatrix}$$

where l_r (resp. l_t) is the distance between the center of the robot (resp. the trailer) and the trailer connection. The inputs of the system are u_1 and u_2 the linear and angular velocities of the robot. To get a basis of $\mathbb{R}^4$ at each configuration $\mathbf{q}$, we define two additional vector fields:

$$\mathbf{X}_3 = \begin{pmatrix} -\sin\theta \\ \cos\theta \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{X}_4 = \begin{pmatrix} -\sin(\theta+\varphi) \\ \cos(\theta+\varphi) \\ -l_t - l_r\cos\varphi \\ -l_t \end{pmatrix}.$$

The linearized system is thus defined by the matrices

$$\bar{A}(s) = \begin{pmatrix} 0 & 0 & -u_1\sin\theta - u_3\cos\theta - u_4\cos(\theta+\varphi) & -u_4\cos(\theta+\varphi) \\ 0 & 0 & u_1\cos\theta - u_3\sin\theta - u_4\sin(\theta+\varphi) & -u_4\sin(\theta+\varphi) \\ 0 & 0 & 0 & u_4 l_r s\varphi \\ 0 & 0 & 0 & \frac{-u_1 c\varphi + u_2 l_r s\varphi}{l_t} \end{pmatrix}$$

$$\bar{B}(s) = \begin{pmatrix} \cos\theta & 0 & -\sin\theta & -\sin(\theta+\varphi) \\ \sin\theta & 0 & \cos\theta & \cos(\theta+\varphi) \\ 0 & 1 & 0 & -l_t - l_r\cos\varphi \\ -\frac{1}{l_t}\sin\varphi & -1 - \frac{l_r}{l_t}\cos\varphi & 0 & -l_t \end{pmatrix}.$$

The input perturbation is defined by truncated Fourier series over inputs u_1 and u_2. The configuration potential field is defined by a decreasing function of the distance to obstacles in the workspace. We refer the reader to [8] for details.

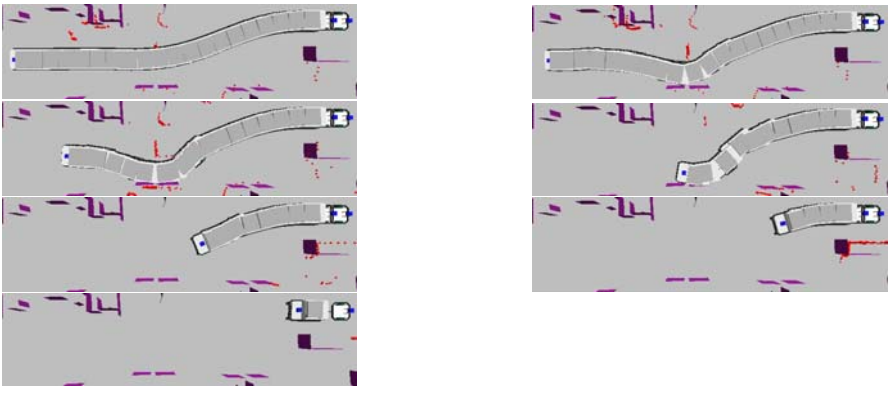


Fig. 17.3 A backward trajectory computed and executed by the mobile robot Hilare 2 towing a trailer. Grey dots are obstacles detected by a laser range finder mounted on the trailer. An unexpected box lies on the trajectory planned by the robot. The robot deforms the trajectory while moving and reach the goal

17.4.1 Experimental Results

Figure 17.3 shows an example where mobile robot Hilare 2 avoids an unexpected obstacle detected by on-board sensors.

17.5 Extension to Docking

The method described in the previous section can be extended to docking of non-holonomic mobile robots by changing the boundary condition relative to the end configuration. This is the topic of this section.

17.5.1 Docking Task

A docking task is a mission given to a robot that consists in following a planned trajectory and reaching a docking configuration. The docking configuration is not defined beforehand as a known robot location, rather it is specified as a set of sensor perceptions from this configuration. The set of landmarks to be perceived when the

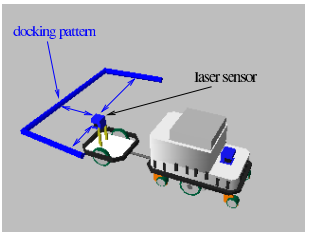


Fig. 17.4 *Docking pattern.* It consists in a set of landmarks defined relatively to a sensor. In this example, the *docking pattern* is defined relatively to the laser sensor mounted on the trailer of a robot

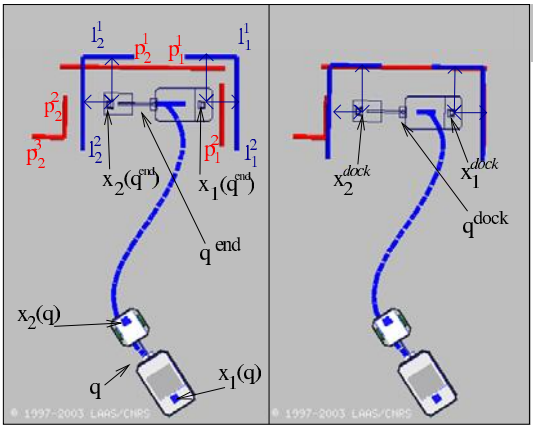


Fig. 17.5 Example of the docking configuration. The robot scans the environment using a 2D laser scanner, from the current configuration along the current trajectory. Extracted straight line segments are matched with the docking pattern to define the docking configuration. The trajectory is progressively deformed in order to make the final configuration tend toward the docking configuration

robot is at the docking configuration is called a *docking pattern*. Figure 17.4 presents such *docking patterns*. On each image, the docking configuration is represented relatively to the docking pattern. Thus a docking task takes as input:

- a collision free trajectory planned within a model of the environment;
- a set of landmarks relative to the docking configuration: the *docking patterns*.

17.5.2 Computation of the Docking Configuration

In the absence of any additional information, the docking configuration is the last configuration of the planned trajectory. Otherwise, the comparison between *docking patterns* and sensor perceptions can be used to compute the docking configuration: *i.e.* the robot configuration where sensor perceptions best match *docking patterns*. We borrow ideas from localization and use a classical extended Kalman filter approach to integrate this information.

17.5.2.1 Probabilistic Framework

The main steps of the computation of the docking configuration are the following.

First, the robot extract features from sensor readings and predicts from the current position of the robot how these features would be seen from the final configuration of the current trajectory. We call those the *predicted features*.

Sensor readings are modeled as Gaussian variables centered on the perfect reading for given robot and landmark positions. The predicted features are matched with the features of the docking pattern using a criterion based on the Mahalanobis distance corresponding to the Gaussian noise associated to the sensors.

The docking pattern is built from a Gaussian random configuration centered on the final configuration of the current trajectory $\mathbf{q}_{rand}$ by evaluating the expected valued conditionally to the predicted features.

Once the docking pattern has been computed, one step of the trajectory deformation algorithm described in Section 17.3 is applied by changing the right hand side of boundary conditions (17.12) by a vector making the final configuration of the current trajectory move toward the docking configuration.

17.6 Experimental Results

We have implemented and tested this method on a real robot. We present the results gathered after experiments in realistic scenarios.

A common scenario for a truck with a trailer is to park the trailer along an unloading platform. That is the final position of the trailer is defined relatively to the unloading platform. We have reproduced this scenario with Hilare 2 towing a trailer.

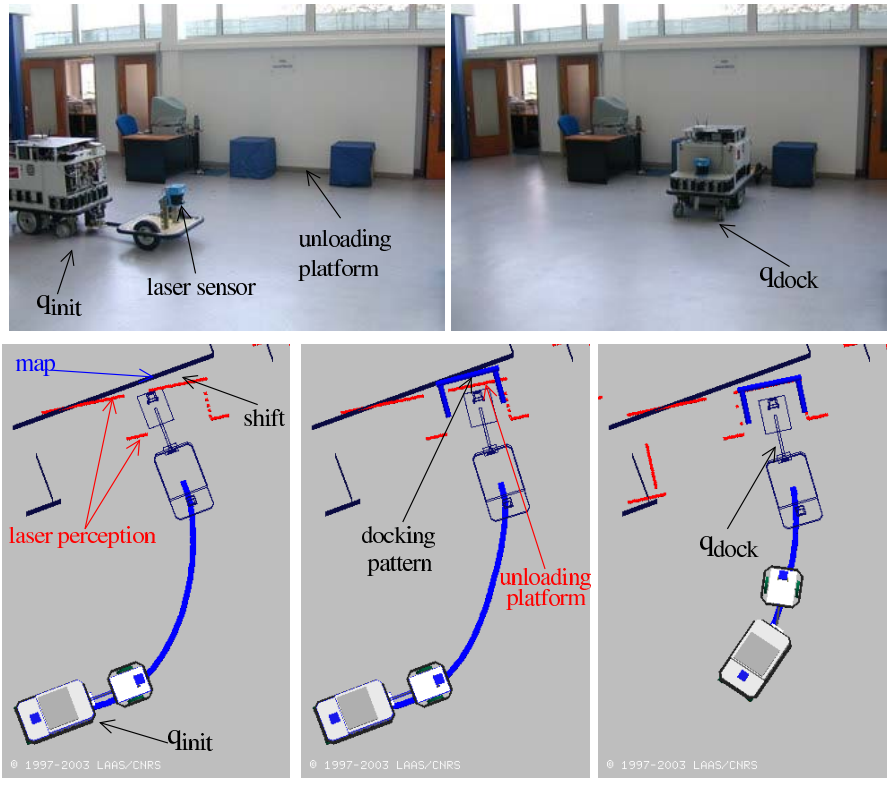


Fig. 17.6 A docking task: the robot is required to reach the unloading platform the shape of which is represented in bold: the *docking pattern*. The laser perception is shifted with respect to the map: it means that the robot is poorly localized. However, the robot is able to detect the *docking pattern* and to deform the reference trajectory in order to avoid obstacles and to dock at the unloading platform. The docking task is executed with respect to the perception and not with respect to the map

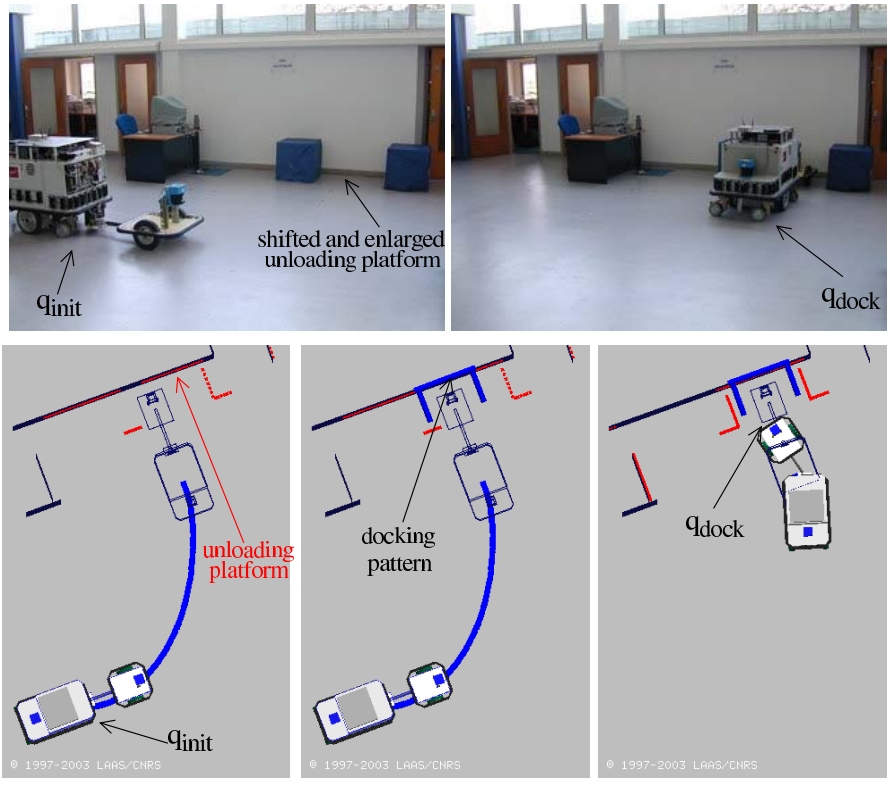


Fig. 17.7 The position and the shape of the unloading platform have been changed compared to figure 17.6. The unloading platform has been shifted to the right and it has been enlarged by 0.2 m. The docking configuration is computed as the configuration where the *docking pattern* best fits the unloading platform

The trailer is equipped with a laser range sensor. In this experiment the landmarks are straight line segments. The *docking pattern* can be composed of any number of segments.

17.6.1 Bad Localization

Figure 17.6 represents this scenario. We see that the map does not perfectly match the perception. This is due to a bad localization of the robot. The docking configuration is anyway computed with respect to the sensor perception. The robot detects the unloading platform. Then it deforms the trajectory in order to dock at the unloading platform and to avoid obstacles. Let us notice that in this experiment the robot does not need to stop to compute the docking configuration nor to deform the trajectory. It is true as long as the docking configuration is close to the end of the trajectory.

17.6.2 The Unloading Platform Has Been Moved

Figure 17.7 illustrates the case where the unloading platform has been moved and the map has not been updated. Moreover, the shape of the unloading platform has changed: it is larger than the *docking pattern*. The matching between the perception and the *docking pattern* is robust to these perturbations and the docking configuration is still defined relatively to the unloading platform.

References

[1] Barraquand, J., Latombe, J.C.: Robot motion planning: A distributed representation approach. International Journal on Robotics Research 10(6), 628–649 (1991)
[2] Brock, O., Khatib, O.: Real-time replanning in high dimensional configuration spaces using sets of homotopic paths. In: International Conference on Robotics and Automation, pp. 550–555. IEEE, San Francisco (2000)
[3] Divelbiss, A., Wen, J.: A path space approach to nonholonomic motion planning in the presence of obstacles. IEEE Transactions on Robotics and Automation 13(3), 443–451 (1997)
[4] Fernandes, C., Gurvits, L., Li, Z.: Optimal Nonholonomic Motion Planning for a Falling Cat. In: Nonholonomic Motion Planning, pp. 379–421. Kluwer Academic Press, Dordrecht (1993)
[5] Fiorini, P., Shiller, Z.: Motion planning in dynamic environments using velocity obstacles. International Journal on Robotics Research 17(7), 760–772 (1998)
[6] Khatib, M., Jaouni, H., Chatila, R., Laumond, J.P.: Dynamic path modification for car-like nonholonomic mobile robots. In: International Conference on Robotics and Automation, pp. 2920–2925. IEEE, Albuquerque (1997)
[7] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. International Journal on Robotics Research 5(1), 90–98 (1986)
[8] Lamiraux, F., Bonnafous, D., Lefebvre, O.: Reactive path deformation for nonholonomic mobile robots. IEEE Transactions on Robotics 20(6), 967–977 (2004)
[9] Lamiraux, F., Sekhavat, S., Laumond, J.P.: Motion planning and control for hilare pulling a trailer. IEEE Transactions on Robotics and Automation 15(4), 640–652 (1999)
[10] Large, F., Sekhavat, S., Shiller, Z., Laugier, C.: Toward real-time global motion planning in a dynamic environment using the nlvo concept. In: International Conference on Intelligent Robots and Systems, pp. 607–612. IEEE/RSJ, Lausanne (2002)
[11] Laumond, J.P.: Controllability of a multibody mobile robot. IEEE Transactions on Robotics and Automation 9(6), 755–763 (1993)
[12] Laumond, J.P., Sekhavat, S., Lamiraux, F.: Guidelines in Nonholonomic Motion Planning for Mobile Robots. In: Robot Motion Planning and Control. LNCIS, pp. 2–53. Springer, NY (1998)
[13] LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. International Journal of Robotics Research 20(5), 378–400 (2001)
[14] Lefebvre, O., Lamiraux, F.: Docking task for nonholonomic mobile robots. In: International Conference on Robotics and Automation, Orlando, USA, pp. 3736–3741 (2006)
[15] Maxim Likhachev, D.F.: Planning long dynamically-feasible maneuvers for autonomous vehicles. In: Proceedings of Robotics: Science and Systems IV, Zurich, Switzerland (2008)

[16] Murray, R.M., Sastry, S.: Steering nonholonomic systems using sinusoids. In: Conference on Decision and Control, pp. 2097–2101. IEEE, Los Alamitos (1990)
[17] Quinlan, S., Khatib, O.: Elastic bands: Connecting path planning and control. In: International Conference on Robotics and Automation, pp. 802–807. IEEE, Atlanta (1993)
[18] Rimon, E., Koditschek, D.: Exact robot navigation using artificial potential functions. IEEE Transactions on Robotics and Automation 8, 501–518 (1992)
[19] vSvestka, P., Overmars, M.: Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In: Proc. IEEE Int. Conf. Robotics and Automation, Japan, pp. 1631–1636 (1995)
[20] vSvestka, P., Overmars, M.: Probabilistic path planning. In: Laumond, J.P. (ed.) Robot Motion Planning and Control. LNCIS, pp. 255–304. Springer, NY (1998)

Chapter 18
Unicycle-like Robots with Eye-in-Hand Monocular Cameras: From PBVS towards IBVS

Daniele Fontanelli, Paolo Salaris, Felipe A.W. Belo, and Antonio Bicchi

Abstract. This chapter presents an introduction to current research devoted to the visual servoing problem of guiding differentially driven robots, more specifically, unicycle-like vehicles, taking into consideration limited field of view (FOV) constraints. The goal is to carry out accurate servoing of the vehicle to a desired posture using only feedback from an on-board camera. First, a position based scheme is proposed, adopting a hybrid control law to cope with limited camera aperture. This scheme relies on a localization method based on extended Kalman filter (EKF) technique that takes into account the robot motion model and odometric data. To increase the potentiality of the visual servoing scheme with respect to existing solutions, which achieve similar goals locally (*i.e.*, when the desired and actual camera views are sufficiently similar), the proposed method visually navigate the robot through an extended visual map before eventually reaching the desired goal. The map construction is part of the approach proposed here, which is then called visual simultaneous localization and mapping (VSLAM) for servoing. Position based scheme accuracy are intrinsically related to the effectiveness of the localization process, which is related to the estimation of 3D information on both the robot and the environment. A shortcut overcoming the estimation process uses visual information directly in the image domain. In this spirit, an image based scheme is presented. The controller is devoted to constantly track desired image feature trajectories. Such trajectories represent optimal (shortest) paths for the vehicle from the 3D initial position towards the desired one. Optimal trajectories satisfies the additional constraint

Daniele Fontanelli
Department of Information Engineering and Computer Science,
University of Trento, Via Sommarive 14, 38123 Povo (TN), Italy
e-mail: fontanelli@disi.unitn.it

Felipe A.W. Belo, Paolo Salaris, and Antonio Bicchi
Department of Electrical Systems and Automation, University of Pisa,
Via Diotisalvi 2, 56100 Pisa, Italy
e-mail: felipebelo@gmail.com, salarispaolo@gmail.com,
bicchi@ing.unipi.it

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 335–360.
springerlink.com

of keeping a feature in sight of the camera and induces a taxonomy of the robot plane of motion into regions. It follows that the robot uses only visual data to determine the region to which it belongs and, hence, the associated optimal path. Similarly to the previous case, the visual scheme effectiveness is improved adopting appearance based image maps.

18.1 Introduction

Vision-based control systems for vehicle control have attracted a lot of attention in the last decades. A plethora of different solutions have been proposed in literature, mainly tailored to the visual architecture of the system at hand. For example, stereo cameras provide accurate environmental measurements with an intrinsic robustness to occlusions. Furthermore, omnidirectional cameras do not suffer of FOV limitations. Despite these advantages, both multiple and omnidirectional cameras need non trivial algorithms to extract useful data from collected images. Moreover, the system mechanical design becomes necessarily complicated and requires accuracy. Hence, it turns out that an economical approach to visual servoing for mobile vehicles is the use of conventional monocular cameras fixed onboard the robot. Nonetheless, the simplest mechanical choice to be used with off-the-shelf conventional cameras, *e.g.*, web cams, reflects in a challenging control design. In fact, mobile robots are usually subject to nonholonomic kinematic constraints and, hence, they cannot rely on independent six-degrees of freedom operational space as robotic manipulators usually rely on. Moreover, when using a limited aperture camera fixed over the robot (a case known as eye-in-hand), the robot must deal with the problem of keeping observed image features within the FOV of the camera while the vehicle maneuvers. Also, the space entities estimate is derived using multiple measurements from different positions in space, taking advantage of multiple-view geometry. This chapter presents an overview of current research devoted to the visual servoing problem of guiding differentially driven robots, with standard eye-in-hand limited aperture monocular cameras, addressing the problems that such a simple and economical framework gives birth to.

In the classic approach to visual servoing, a well known taxonomy was derived and classified in the seminal paper of Weiss in [31]: in image-based visual servoing (IBVS) the control error is defined directly in the image space, based on *features* extracted from image data, *e.g.*, visual cues like points, planes or lines; on the other hand, position-based visual servoing (PBVS) computes the error in relation to a set of 3D parameters that are estimated from image measurements, *e.g.*, robot position errors with respect to the desired position to reach. In the second case, position errors are usually computed in the robot Cartesian space and provided, as customary, to the control system. Robot position reconstruction is often referred to as robot *localization*. The two vision-based schemes thus described should be regarded as the end-points of a range of different possibilities, whereby the raw sensorial information is gradually abstracted away to a more structured representation using some knowledge of the robot-environment model.

While PBVS methods are the most straightforward ones, they require metrical information about the feature positions, usually organized in a *map*. When the metrical information can't be computed directly from the camera view (which is the case for a monocular camera), pose estimation is usually achieved with the additional use of external measurements, such as odometric data. Therefore, the latest research focuses on *hybrid* methods or pure IBVS methods to overcome the estimation problem. Among the others, hybrid schemes have been proposed in [11], coming up to a scheme which is roughly half-way between IBVS and PBVS, and in [16], where a *switched* control approach that utilizes both schemes depending on the distance to the target has been implemented. Indeed, IBVS is more accurate than PBVS as soon as the reference and the target images are close enough ([4, 5]). For this reason, an image based task is often separated into a set of consecutive control problems using a sequence of target images, in other words using appearance-based visual maps that not take into account any 3D spatial information ([28, 15]).

In the past few years, robot scientists have focused on the optimality of paths followed by visually servoed robots. For example, researchers have focused on the optimal control of visually guided robotic manipulators ([8]) or on optimal trajectory planning for robot manipulators controlled via a limited FOV camera ([13]). Minimal trajectories have been also presented in [25] in case of large displacements, again for a six degrees of freedom robot manipulator. Optimal paths for differentially driven robots with visibility restricted to limited FOV have recently been addressed by some researchers ([2, 30]). The solutions proposed in this field are more related to optimal path planning than robot reactive control, restricting the role of the visual control to path following. Of course, also in this case solutions can be divided in position based or image based, depending on the space in which the trajectories are derived.

In this chapter we aim to provide an overview to the visual servoing of differentially driven robot and the vision theoretic fundamentals needed in each case. Then, a PBVS control scheme that solves the problem in the 3D domain is proposed, together with a servoing-oriented simultaneous localization and mapping (SLAM) algorithm to enhance the potentiality of the controller. To overcome the localization process, a combination of an IBVS and an optimal (shortest) path planner is then proposed. Again, the autonomous capability of the servoed robot are increased adopting an appearance map-based approach. A discussion on the advantages and drawbacks that pertain to each technique is also presented.

18.2 Problem Definition

The visual servoing problem as meant in this chapter is referred to mobile vehicles, with a rigidly fixed on-board camera. In particular, we consider vehicles that constantly move on a plane, as in typical indoor set-up, like factory or office floors. We assume for the moving platform a driftless kinematic model, more precisely this chapter refers to a *unicycle-like* nonholonomic mobile robot. Without loss of generality, we assume that the robot coordinates are measured with respect to a dextrous

reference frame $\langle W\rangle = \{O_w, X_w, Y_w, Z_w\}$, fixed with the static environment. It is also assumed that the Y_w coordinate $y(t) = 0$, $\forall t$, hence the state space of the mobile platform is $\xi(t) = (x(t), z(t), \theta(t))$, where the robot reference point $(x(t), z(t))$ is in the middle of the wheel axle and the robot direction $\theta(t)$ is zero when the vehicle heads to the X_w axis. Assuming that the control inputs are $u(t) = (v(t), \omega(t))$, with $v(t)$ and $\omega(t)$ are respectively the forward and angular velocities of the vehicle, the system kinematic model is

$$\dot{\xi} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega = f_v v + f_\omega \omega. \tag{18.1}$$

The mobile agent is equipped with a rigidly fixed camera with a reference frame $\langle C\rangle = \{O_c, X_c, Y_c, Z_c\}$ such that the optical center O_c corresponds to the robot's center and the optical axis Z_c is aligned with the robot's forward direction. If the robot orientation is null, the Z_c axis is parallel to the X_w axis, with the same direction, and the X_c axis is parallel to the Z_w axis, with opposite direction.

Motionless features will be defined in reference to the camera frame as ${}^cP = [{}^cx, {}^cy, {}^cz]^T$ and to the fixed frame $\langle W\rangle$ as ${}^wP$. The coordinate transformation between $\langle W\rangle$ and $\langle C\rangle$ is given by $[{}^cP^T, 1]^T = [{}^cx, {}^cy, {}^cz, 1]^T = {}^cH_w[{}^wP^T, 1]^T$, where ${}^cH_w$ is the transformation matrix between frames. Assume a *pinhole camera model* where α_x and α_y are the focal lengths of the intrinsic camera calibration matrix ([18])

$$K_c = \begin{bmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{18.2}$$

The origin O_I of the image plane reference frame $\langle I\rangle = \{O_I, X_I, Y_I\}$ is assumed to be coincident with the principal point, *i.e.* the intersection of the camera axis Z_c with the image plane. The feature coordinates in the image frame (measured in pixels) are ${}^Ip = [{}^Ix, {}^Iy]^T$. Hence, the projective transformation (or *homography*) that describes a general mapping from ${}^wP$ to ${}^Ip$ can be summarized as

$$\lambda\, {}^I\bar{p} = K_c[I_3|\mathbf{0}^T]\, {}^cH_w\, {}^w\bar{P}, \tag{18.3}$$

where λ is an homogeneous scale factor, and ${}^I\bar{p}$ and ${}^w\bar{P}$ are the homogeneous representation of vectors ${}^Ip$ and ${}^wP$.

With this assumption, the image feature velocities are derived using (18.1) and (18.2), usually referred to as *image Jacobian*

$${}^I\dot{p} = \begin{bmatrix} {}^I\dot{x} \\ {}^I\dot{y} \end{bmatrix} = \begin{bmatrix} \frac{{}^Ix\,{}^Iy}{\alpha_y\, {}^wy} & \frac{{}^Ix^2+\alpha_x^2}{\alpha_x} \\ \frac{{}^Iy^2}{\alpha_y\, {}^wy} & \frac{{}^Ix\,{}^Iy}{\alpha_x} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{18.4}$$

In the visual servoing literature, whenever an eye-in-hand configuration is considered (as is a camera rigidly fixed on a moving platform), the objective of the

control task is to stabilize the robot towards the desired position controlling the camera position ([4, 5, 23]). More precisely:

Definition 18.1. *Given the desired and the current robot positions, which correspond the desired* $\langle C\rangle_d = \{O_{cd}, X_{cd}, Y_{cd}, Z_{cd}\}$ *and the current* $\langle C\rangle_c = \{O_{cc}, X_{cc}, Y_{cc}, Z_{cc}\}$ *reference frames respectively, the stabilization in the desired position is accomplished if* $\langle C\rangle_c \equiv \langle C\rangle_d$ *at the end of the control task.*

Remark 18.1. Since the mobile robot is moving constantly on a plane, the visual servoing control approach is used only to stabilize at most a 3D subspace of the state space $\xi \in \mathbb{R}^n$.

As it is customary in the visual servoing literature, by a suitable robot state variables change of coordinates, $\langle W\rangle \equiv \langle C\rangle_d$. Hence, the visual servoing control problem turns into a *point-to-point stabilization* problem, *i.e.*, we require that $\xi(t) \to 0$ as $t \to +\infty$. In the particular framework proposed in this chapter, $\langle W\rangle \equiv \langle C\rangle_d$ corresponds to choose $X_w = Z_{cd}$, $Y_w = Y_{cd}$ and $Z_w = -X_{cd}$.

A visual servoing scheme for robot control relays on the straightforward reformulation of Definition 18.1 from the use of image feature positions (albeit the equivalence between the two problems holds only if singularity configurations are avoided, see [4, 26]).

Definition 18.2. *Given n current* $F_c = [{}^I x_{c_1}, {}^I y_{c_1}, {}^I x_{c_2}, \ldots, {}^I y_{c_n}]^T$ image feature *positions, the servoing task is fulfilled if at the end of the controlled trajectory,* F_c *matches the desired image feature positions* $F_d = [{}^I x_{d_1}, {}^I y_{d_1}, {}^I x_{d_2}, \ldots, {}^I y_{d_n}]^T$ *, i.e.,* ${}^I x_{d_i} = {}^I x_{c_i}$ *and* ${}^I y_{d_i} = {}^I y_{c_i}$, $\forall i = 1, \ldots, n$.

In the presented chapter, we consider the visual servoing with an explicit feasibility constraint: the image features must be always within the FOV of the camera along the robot stabilizing trajectories (henceforth referred to as the *FOV constraint*), which ensures that a visual feedback can be always performed. Probably, the problem of keeping the features in view during the robot manoeuvres is one the most relevant problem to address for effective robot control. Multiple solutions have been proposed in literature, ranging from omnidirectional cameras ([1]), image path planning ([9]), or switching visual servoing schemes ([7]). In the presented dissertation, the FOV constraint will be addressed in the controller design, which simplifies the mechanical set-up and lowers the overall cost. Moreover, we will not focus on image processing backgrounds, giving for granted the feature extraction, tracking and association among the image features. In particular, we adopt the well-established scale invariant feature transform (SIFT) proposed in [21].

18.2.1 Position-based Visual Servoing

The main attractive feature of the PBVS approach is probably the relative simplicity of the control design. Indeed, the control law can be synthesized in the usual working coordinates for the robot ([6]). Unfortunately, a position estimation algorithm has to be necessarily provided. More precisely, since the estimated robot posture

$\hat{\xi}(t)$ can be derived by the knowledge of ${}^c\hat{H}_w(t)$ (at least for the subspace of interest), the objective of the estimation algorithm, often dubbed *localization* or *pose estimation* algorithm, is to reconstruct the relative position between $\langle C \rangle_c$ and $\langle W \rangle$.

Notice that if ${}^c\hat{H}_w(t)$ is composed of a rotation R and a translation T, (18.3) is rewritten as $\lambda {}^I\bar{p} = K_c[R|T]{}^w\bar{P}$. The fundamental matrix F associated with this projective map is defined as $F = K_c^T S(T) R K_c^{-1}$, where $S(T)$ is the skew-symmetric matrix associated with the translation T. Trivially, if ${}^I p_d$ is the feature position in the desired camera position, we have $\lambda {}^I\bar{p}_d = K_c[\mathbf{I}|0]{}^w\bar{P}$, with the fundamental matrix F satisfying the condition ${}^I\bar{p}_d F {}^I\bar{p} = 0$. Therefore, assuming a calibrated camera (K_c is known), the fundamental matrix can be robustly estimated knowing image features correspondences ${}^I p \leftrightarrow {}^I p_d$ ([18]). Therefore, two main design requirements are derived:

1. full camera calibration (as implicitly assumed in the rest of the chapter);
2. a priori knowledge of ${}^w P$ for each feature.

The latter condition is not strictly needed if the algorithm that performs the servoing works in agreement with a mapping algorithm. In this case, the visual servoing scheme can be applied to previously unknown portion of the environment, thus increasing the autonomous capability of the mobile vehicle. The VSLAM for servoing is an example of such an architecture. In fact, since the PBVS approach abstracts sensor information to a higher level of representation, it allows the integration of different sensorial sources, thus make it suitable to cooperate with SLAM based architectures. In our example of a camera mounted on a mobile robot, for instance, the synergistic use of odometry and visual feedback is viable if these information are described in the same coordinate frame, where they can be fused coherently.

Summarizing, the PBVS approach estimates the error between the current and desired robot position through an approximation of ${}^c\hat{H}_w(t)$. For this reason, we can conclude that the PBVS approach verifies the Definition 18.1, which implies the satisfaction of Definition 18.2 in accordance with the accuracy of the estimation algorithm.

18.2.2 Image-based Visual Servoing

IBVS and other sensor-level control schemes have several advantages in relation to PBVS, such as robustness (or even insensitivity) to modelling errors, and hence suitability to unstructured scenes and environments ([24]). Although IBVS is demonstrated to be quite effective for manipulators ([28]), its control design turns out to be more challenging for nonholonomic mobile vehicles. Indeed, the *image Jacobian* (18.4), which relates image features and robot's motion, cannot be used to solve the general stabilization problem directly, a fact reported in famous results on nonholonomic systems stabilization ([3]). However, as presented here, the classical visual servoing scheme can be used together with path planners on image maps ([4]). Such maps can be either given a priori or constructed on-line, according to a VSLAM approach.

For IBVS the image error between two different postures is computed directly on the image measurements. Hence, an IBVS controller relates to Definition 18.2, which implies the satisfaction of Definition 18.1 if pathological image feature postures (singularities) are avoided.

18.2.3 Optimal Trajectories

Path planning solutions to vision-based parking have been proposed in literature ([23]). Among all the possible choices for robot trajectories, a particular mention deserves those one that are optimal, *e.g.*, minimum length. Moreover, optimal path planning turns to be particularly challenging when the nonholonomicity of the platform combines with the FOV constraint. A solution to this problem has been provided very recently by [2], where shortest paths are shown to be comprised of three maneuvers: a rotation on the spot, a straight line and a logarithmic spiral. The results there proposed are additionally refined in [30], where a global partition of the motion plane in terms of optimal trajectories has been derived.

As the optimal trajectories are retrieved, a visual servoing scheme can be used to control the robot along the optimal trajectories. For example, [20] proposes a PBVS homography-based controller. Alternatively, IBVS approaches can be used, as reported in what follows.

18.3 PBVS in the Large

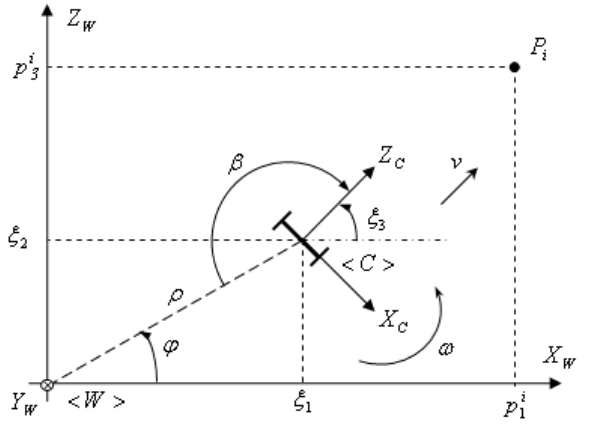
As described previously, the PBVS comprises two major components that are described briefly in what follows: a localization algorithm and the controller design. The quantities referred in this section are reported in Figure 18.1(a), where the fixed frame $\langle W \rangle$ and the camera frame $\langle C \rangle$ are reported.

18.3.1 Robot Localization

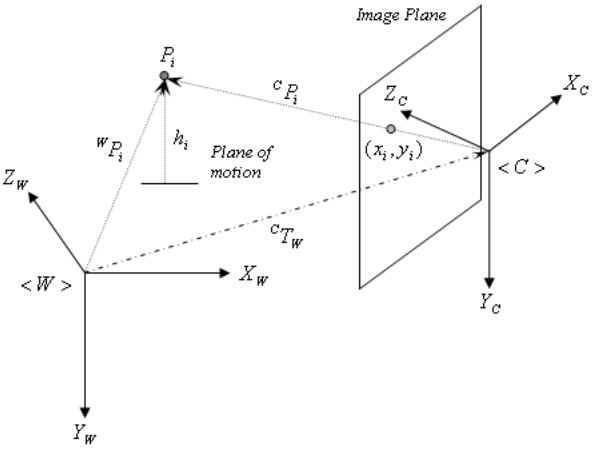
Let ${}^w\xi = [\xi_1, \xi_2, \xi_3]^T$ be the set of Cartesian coordinates of the robot (see Figure 18.1(a)). The current position of the feature ${}^cP_i$ in the camera frame is related to ${}^wP_i$ in the fixed frame by a rigid-body motion ${}^cH_w$, which can be computed assuming the knowledge of the height of the features ${}^cy_i$, $\forall i$. Indeed, with respect to Figure 18.1(b), we have

$$\begin{bmatrix} {}^cx_i \\ {}^cz_i \end{bmatrix} = \begin{bmatrix} {}^wz_i & {}^wx_i & 1 & 0 \\ -{}^wx_i & {}^wz_i & 0 & 1 \end{bmatrix} b, \tag{18.5}$$

with $b = [-\cos\xi_3, \sin\xi_3, \xi_2\cos\xi_3 - \xi_1\sin\xi_3, -\xi_1\cos\xi_3 - \xi_2\sin\xi_3]^T$. Equation 18.5 can be regarded as providing two nonlinear scalar equations in the 3 unknowns (ξ_1, ξ_2, ξ_3), for each feature observed in the current and desired images. Assuming a number $n \geq 4$ of features, the actual unknown position and orientation $\xi(t)$ of the unicycle can be evaluated by solving for b in a least-squares sense (see [26] for further details).



(a)



(b)

Fig. 18.1 (a) fixed frame $\langle W \rangle$, camera frame $\langle C \rangle$, and relative coordinates (ξ_1, ξ_2, ξ_3) and (ρ, ϕ, β); and (b) fixed frame $\langle W \rangle$, camera frame $\langle C \rangle$ and relative feature coordinates

The localization method thus presented is based on punctual estimation and does not take into account the known robot motion model. To better exploit the available information, an EKF is adopted, given for granted the association between current and desired point features. However, should feature outliers occur in the process, more robust filtering should be used in place of simple EKF, such as *e.g.* those described in [32] or [22].

The EKF localization state is $S = [\xi_1, \xi_2, \xi_3, {}^cx_1, \ldots, {}^cx_n, {}^cz_n]^T$. Estimated state initial guess is computed using the previously presented least mean squares static localization. The initial model covariance matrix P_0 is block diagonal. $P_0^r \in \mathbb{R}^{3\times3}$,

i.e., initial covariance matrix for the vehicle, has its values set to half a meter for ξ_1 and ξ_2, while it is one radian for ξ_3. $P_0^i \in \mathbb{R}^{2\times 2}$, *i.e.*, the block diagonal matrices for the features, have their entries set to ≈ 20 centimeters in our experimental setting. Unicycle kinematic model, with odometric data, is assumed for state prediction.

Two different noise sources are taken into account. The prediction errors are modelled as additive and zero mean Gaussian noises with covariance matrix $Q = P_0$. Systematic errors are assumed to be removed by suitable calibration, hence nonzero mean errors are not modelled. The odometry errors γ_r, γ_l, for the right and left wheel respectively, are assumed to be zero mean and Gaussian distributed. They are also assumed to be equal for both wheels. This is a simple but easily verified assumption in respect to generic unicycle like vehicles, and is computed taking into account lack of accuracy in odometry (typically due to wheels slipping and skidding). The covariance matrix of the prior estimate (model prediction) is then calculated by the formula

$$P_k^- = A_k P_{k-1} A_k^T + \sigma_\gamma^2 B_k B_k^T + Q,$$

where σ_γ^2 is the input variance ($\gamma = \gamma_r = \gamma_l$ by assumption), A_k and B_k are the model Jacobians and P_{k-1} is the model covariance matrix at the previous step.

During experiments, the EKF-based localization thus derived shows to outperform the least mean squares approach, as it may be expected.

18.3.2 Visual Servoing with Omnidirectional Sight

Consider that only one feature has to be tracked, coincident with the origin O_w. Consider for this problem a new set of coordinates, which is better suited to describe the angle by which the feature is observed from the vehicle, described by $\Phi : \mathbb{R}^2 \times S \to \mathbb{R}^+ \times S^2$ (see Figure 18.1(a)) and the new dynamics

$$\begin{bmatrix}\rho\\ \phi\\ \beta\end{bmatrix} = \begin{bmatrix}\sqrt{{\xi_1}^2 + {\xi_2}^2}\\ \arctan(\frac{\xi_2}{\xi_1})\\ \pi + \arctan(\frac{\xi_2}{\xi_1}) - \xi_3\end{bmatrix}, \text{ and } \begin{bmatrix}\dot\rho\\ \dot\phi\\ \dot\beta\end{bmatrix} = \begin{bmatrix}-\rho\cos\beta & 0\\ \sin\beta & 0\\ \sin\beta & -1\end{bmatrix}\begin{bmatrix}u\\ \omega\end{bmatrix}, \tag{18.6}$$

where we let $u = \frac{v}{\rho}$. Observe that this change of coordinates is a diffeomorphism everywhere except at the origin of the plane (exactly where the feature point is). A continuous, time-invariant control law can in principle stabilize the system (18.6). Indeed, the two control vector fields are now linearly dependent at the origin, thus making Brockett's negative result [3] unapplicable.

Consider the candidate Lyapunov function $V = \frac{1}{2}(\rho^2 + \phi^2 + \lambda\beta^2)$, with $\lambda > 0$ a free parameter to be used in the following controller design. Substituting

$$u = \cos\beta, \text{ and } \omega = \frac{\phi\sin\beta\cos\beta + \lambda\beta\sin\beta\cos\beta}{\lambda\beta} + \beta \tag{18.7}$$

in the Lypunov derivative one gets $\dot{V} = -\rho^2 \cos^2\beta - \lambda\beta^2 \leq 0$. By using LaSalle's invariant set theorem, the controlled dynamics obtained plugging (18.7) in (18.6) turns to be asymptotically stable.

18.3.3 Visual Servoing with FOV Constraint

The basic idea to be applied in this section is rather simple, and is based on the fact that the Lyapunov-based control described in the previous section is not uniquely defined. Rather, a whole family of controllers can be defined by simply redefining the control Lyapunov function candidate. It can be expected that for such different candidates, the resulting stabilizing control laws and ensuing trajectories are different, and that switching among these control laws should be enabled when the FOV constraint for the *i*-th feature

$$\gamma(\rho,\phi,\beta) = \phi - \pi - \beta - \arctan\frac{{}^{w}z_i - \rho\sin\phi}{{}^{w}x_i - \rho\cos\phi} = \arctan\frac{{}^{I}x_i}{\alpha_x} \in [-\Delta,\Delta] \qquad (18.8)$$

is about to be violated (see [26] for further details). Notice that the limited FOV is described by a symmetric cone centered in the optical axis Z_c with semi-aperture Δ.

The switching controller is expressed in a set of different polar coordinates, which are conveniently denoted by introducing the two vectors $\tilde{\beta} = [\beta, \beta-\pi, \beta-\pi, \beta+\pi, \beta+\pi, \beta+\pi]$ and $\tilde{\phi} = [\phi-\pi, \phi-2\pi, \phi, \phi-2\pi, \phi]$. Correspondingly, a set of five distinct candidate Lyapunov functions can be written as $V_i(\rho,\alpha,\beta) = \frac{1}{2}(\rho^2 + \tilde{\phi}_i^2 + \tilde{\beta}_i^2)$, with $i = 1,\ldots,5$. The control law choice, *i.e.*

$$u = \cos\beta, \text{ and } \omega = \lambda\tilde{\beta}_i + \frac{\sin\beta\cos\beta}{\tilde{\beta}_i}(\tilde{\phi}_i + \tilde{\beta}_i),$$

is such that all the Lyapunov candidates have negative semi-definite time derivatives and (by LaSalle's invariant set principle) asymptotically stable. These five different control laws (parameterized by λ) define in turn five different controlled dynamics (analogous to (18.6)) that are globally asymptotically stable in the state manifold $\mathbb{R}^+ \times S^2$. Although none of these control laws alone can guarantee that the FOV constraint is satisfied throughout the parking maneuver, it is shown that a suitable switching logic among the control laws achieves this goal. The switching law is triggered when, during the stabilization with one of the five control laws, a feature approaches the border of the FOV by a threshold $\Delta_j < \Delta$, *i.e.* when $|\gamma| \geq \Delta_j$. It should be noticed that a dead zone is introduced in the controller for $\rho \leq \rho_D$, within which the forward velocity control u is set to zero and the so-called Zeno phenomenon is avoided.

18.3.4 Visual Servoing in the Large

The classic approach for visual servoing, however, so far have focussed on local stabilization, in the sense that the initial and desired conditions of the system are

assumed to be close enough so that a significant number of features remain in view all along the maneuver. The purpose of this section is to define the tools that enables its use to servo the vehicle *in the large*, *i.e.* across paths connecting totally different initial and final views. The necessary information are stored in images, called *way-points*, which can be used to topologically connect the initial and desired images, and in a *metric* map, which stores sufficient data to implement the PBVS. A representation of the environment that conveys these metric and topological information will be referred to as a *hybrid visual map*.

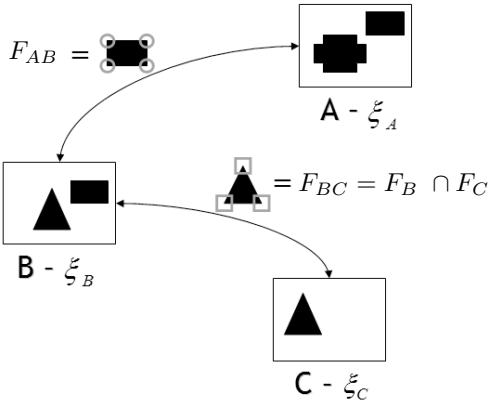
The literature on VSLAM is rather extensive (see *e.g.* [27, 10, 12, 29]), and COTS software is already available ([19]). These results are clearly fundamental for the approach to servoing here presented, which instead relates to nonholonomic vehicles. In the hybrid map representation, the metric information is represented by a set of robot postures, along with the corresponding 3D position estimates for the features observed from such postures. The topological information is represented by an undirected reachability graph (we assume indeed that possible environment changes do not affect the traversability of the space by the robot, [14]). The hybrid map construction method is described in what follows:

1. from the initial unknown position of the vehicle (*i.e.* ${}^W\xi_A = {}^W[0,0,0]^T$) an image I_A of a portion of the scene in view is grabbed and stored in the first node A of the hybrid map (see Figure 18.2(a)));
2. from the image in view, a subset F_A of n_A features is selected;
3. the vehicle moves, avoiding obstacles with proximity sensors, in an arbitrary direction using a simple control law that keeps the image point features in view;
4. an EKF is implemented using odometry and camera measurements to estimate the relative spatial position of the feature in camera frame $\langle C \rangle$;
5. once 3D feature position estimates have converged to a value under given level of uncertainty determined by the covariance matrix, the robot stops moving, updates the metric map (Figure 18.2(b)) and then a new node corresponding to the current pose is added to the hybrid map;
6. to add new nodes from the already created ones, the procedure starts again from step 2.

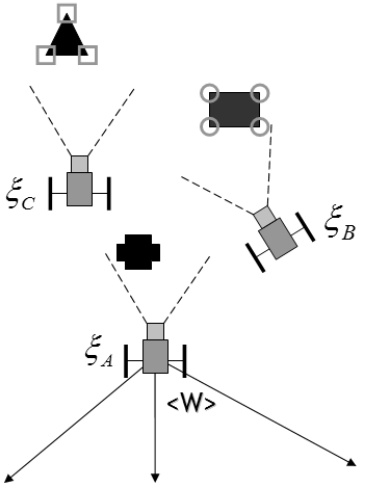
Let the robot be in a generic mapped position, say ${}^W\xi_A = {}^W[\xi_1, \xi_2, \xi_3]^T$ (or a node A with image I_A). Suppose that the robot has to reach a new position, say ξ_K, expressed in the metric map. If ${}^W\xi_K$ corresponds to a topologically mapped location K, which has an associated image I_K (found using SIFT technique), a standard graph visiting algorithm is used for the path selection from A to K in the image map, therefore, permitting the vehicle to steer through the map nodes using the servoing presented previously. For space limits, we refer the interested readers to [14] for further details on the mapping/navigation processes here briefly discussed.

18.3.5 Experimental Results

Two experiments, with different experimental set-up are reported. The first experimental setup comprises of a TRC LabMate vehicle, equipped with an analogical



(a)



(b)

Fig. 18.2 (a) topological map: grabbed images are indicated with a capital letter, say I_A, I_B, I_C. Each grabbed image corresponds to both a certain robot configuration in the metric map (b) and a node in the topological map. The nodes A and B are connected if and only if the set of features $F_{AB} = F_A \cap F_B$ is not empty; and (b) metric map: the positions A, B, C, representing each image node in the topological map (a), ξ_A, ξ_B and ξ_C are a set of 3D robot postures

monochromatic camera Jai CVM-50 placed on the robot so that a vertical axis through the camera pinhole intersects the wheel axis in the midpoint. The camera allows for a $\Delta = \pi/6$ semi-aperture of the optical cone, while a threshold of $\Delta_j = \pi/8$ was used in the experiment to switch among different controllers. The controller is

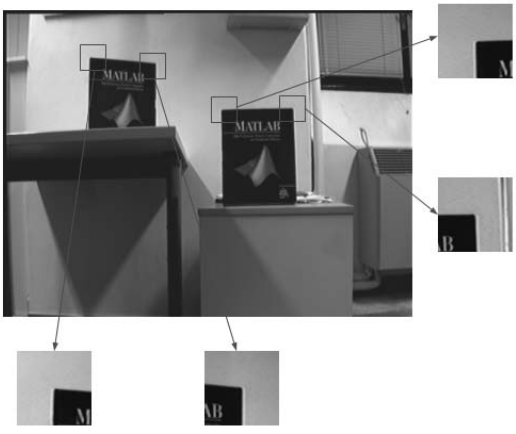


Fig. 18.3 Image grabbed from the target position, with the four selected control features shown in detail

implemented under Linux on a 300MHz PentiumII[1] PC equipped with a Matrox[2] MeteorI frame grabber. The XVision library is used to compute optical flow and to track features. The hardware communication between the robot and the PC is performed by a RS-232 serial cable. In this experiment, the PBVS controller alone is shown. Four features from the scene are used as the desired feature set, belonging to the desired image, recorded in a preliminary phase of the experiment (see Figure 18.3). Images grabbed from the robot camera in the initial, offset configuration and at the end of the visually-servoed parking maneuver are shown in Figure 18.4, along with ground-reference views showing the experimental environment.

In the second experimental set-up, a low-cost apparatus was employed to highlight the potential of the proposed technique. The experimental setup is comprised of a K-Team Koala vehicle, equipped with a commercial web-cam placed on the front part of the robot platform. The controller is implemented under Windows XP on a 1130MHz Pentium III laptop mounted on-board. SIFT elaboration is performed using evolution robotics software platform (ERSP) vision library (see [17, 19]). The Intel OpenCV library was used to compute optical flow and to track features. The hardware communication between the robot and the laptop is again performed by a RS-232 serial cable.

In the experiments both the mapping phase (topological and metric) and the navigation phase (visual servoing) are reported. In the mapping experiment, the robot

[1] Pentium™ is a trademark of Intel Corporation in the U.S. and other countries. http://www.intel.com.

[2] Matrox and the names of Matrox products referenced herein are either trademarks and/or service marks or registered trademarks and/or service marks of Matrox. http://www.matrox.com

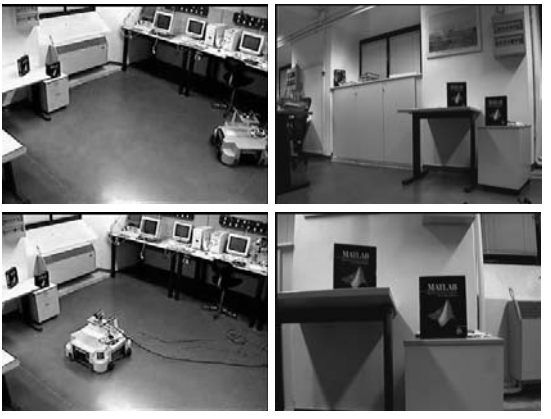


Fig. 18.4 External views (left column) and subjective images (right column) as taken from the vehicle, in the initial configuration (top row) and in the final configuration (bottom row), after reaching convergence under the proposed visual feedback control scheme. The bottom right image should be compared with the target image in Figure 18.3

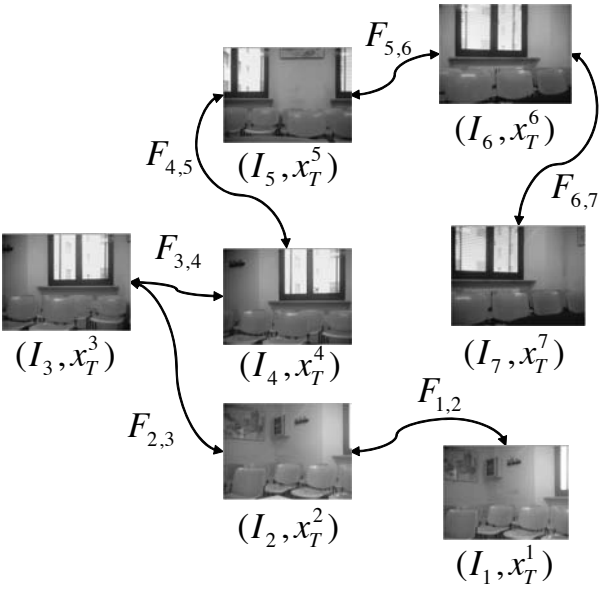


Fig. 18.5 The topological image-based map

collects a set of images and, for each pair of images, say I_i and I_j, it estimates the 3D coordinates of the image feature points F_{ij} and the 3D robot positions ξ_i and ξ_j.

The experimental mapping process runs for 50 seconds and the sampling period (*i.e.* the inverse of the frequency of the EKF steps) is $T = 0.1$ seconds. The sampling period T is determined by the worst-case frame rate available for commercial

(a) (b)

Fig. 18.6 Map navigation: (a) desired image; and (b) initial image

Fig. 18.7 Desired images from each topological map node on the top row. Images grabbed from the camera after each visually servoed path, on the bottom row

web-cams. Although even low cost cameras ensure about 20 to 30 frames per second, the rate changes depending on ambient illumination variations.

Figure 18.5 reports an image graph created during the topological mapping phase.

The visual servoing controller has been used to travel the distance between the mapped images, parking the vehicle in the position x_T^3 (position I_3 in the topological map, see Figure 18.6(a)). The initial robot position is unknown, but the architecture solves the kidnapped robot problem identifying the topological position x_T^5 (see Figure 18.6(b)). Hence, the visual servoing path corresponds to a travel between image node I_5 to I_3.

In Figure 18.7, the nodes crossed by the robot during the parking are represented. In the top row, the images stored in the topological map (*i.e.* desired images for the visual servoing) are represented, while in the bottom row, the images grabbed from the camera after each path are depicted. Approaching an intermediate node ends once it is possible to localize and to track features of the next node to be reached.

A wide movement in the mapped environment comprises several limited movements between each pair of images (Figure 18.7). Nevertheless, the visual servoed motion between successive images is still quite small. Indeed, it is well known in

the literature (*e.g.* [4, 5]) that large image errors (hence, large robot movements) decrease accuracy and robustness of the visual servo controller. In the proposed architecture, the granularity of the topological map is then related to the visual servoing accuracy.

18.4 Optimal Trajectories

This section presents a solution to the path planning problem of a unicycle-like vehicle subject to limited FOV constraint. In particular, the paths derived are the minimum length (optimal) paths between initial and desired positions.

Assume that the tracked feature position is coincident with the origin of the reference frame O_w and that the total horizontal aperture of the camera's optical cone is given by $\delta = 2\Delta$. The desired position P of the robot on the plane of motion w.r.t. $\langle W \rangle$ is assumed to lie on the X_w axis, with polar coordinates $(\rho_P, 0)$. Hence, the desired state of the robot is $(\rho_P, 0, \pi)$ with respect to the model (18.6). The objective of the optimal trajectory problem with FOV constraint is the partition of the motion plane into regions, depending on the desired position P. From all initial positions Q inside a region, the structure of the shortest paths turns to be invariant.

Denote with "$*$" a zero-length rotation on the spot, with SL a straight line and with T a logarithmic spiral. The three different kinds of maneuvers that compose the optimal paths, according to [2, 30], are then symbolically described. While the mathematic description of $*$ and SL is straightforward, logarithmic spirals deserve some additional explanations. In general, the equation of the logarithmic spiral that passes through a point Q (whose polar coordinates are (ρ_Q, ψ_Q) w.r.t. $\langle W \rangle$) is expressed as $T_Q : \left(\rho_Q e^{(\psi_Q - \psi)g}, \psi\right)$, where $g = \frac{\cos\alpha}{\sin\alpha}$ and α is the spiral's characteristic angle. The spiral rotates clockwise if $\alpha < 0$ (denoted with $T1_Q$) and counterclockwise if $\alpha > 0$ (denoted with $T2_Q$). Notice that if $\alpha = 0$, the logarithmic spiral is a straight line passing through the origin O_W, and, if $\alpha = \pm\pi/2$, it is a circumference.

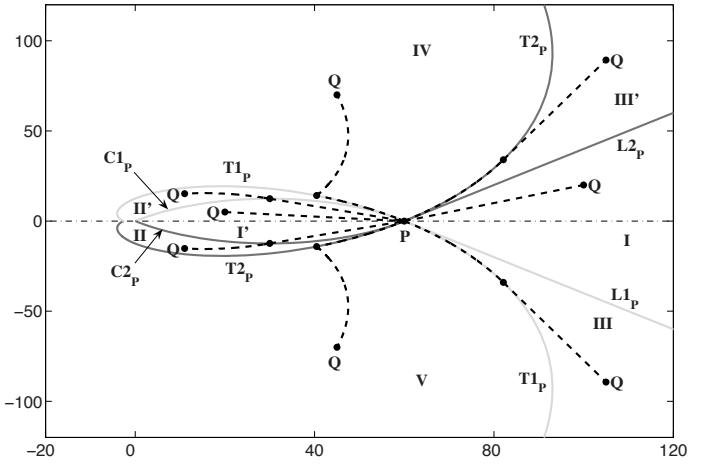
Using standard tools from optimal control theory, it can be shown that optimal words that minimizes the total length of the path

$$\mathscr{L} = \int_0^T |v|\, dt\,,$$

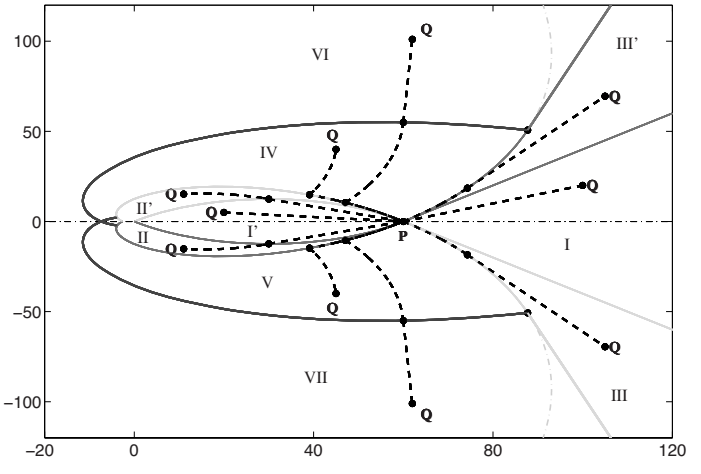
are words of extremal arcs that can be covered forward or backward w.r.t. the direction of motion. Summarizing, extremal paths will be characterized by sequences of symbols $\{*, SL, T1, T2\}$.

Let us denote the two logarithmic spirals that pass through the desired position P as $T1_P$ and $T2_P$, with characteristic angles $\alpha = -\Delta$ and $\alpha = \Delta$, respectively. The two spirals divide the plane into four disjoint regions. Using the geometric properties of the problem, these regions are further subdivided considering the set of points Q for which the optimal path is given by a straight-line from Q to P without violating the FOV constraints ([2]). With this intuitive subdivision, eight regions are derived, depicted in Figure 18.8(a). Defining a smooth transition between segments with the symbol "–", the optimal path from each region is defined as follows:

- (*Region I*) region between the half-lines $L1_P$ and $L2_P$. Shortest path: SL;
- (*Region I'*) region between the two arc circles $C1_P$ and $C2_P$. Shortest path: SL;
- (*Region II*) region between the circle arc $C2_P$ (bound of *Region I'*) and the spiral $T2_P$ (bound of *Region V*), symmetric to *Region II'*. Shortest path: $T2 - SL$;
- (*Region III*) region between the half-line $L1_P$ (bound of *Region I*) and the spiral $T1_P$ (bound of *Region V*), symmetric to *Region III'*. Shortest path: $SL - T1_P$;



(a)



(b)

Fig. 18.8 Shortest paths: (a) according to [2]; and (b) according to [30]

- (*Region IV*) region between the spiral $T2_P$ (bound of *Region III'*) and the horizontal line passing through the feature and the desired position P (bound of *Region V*). The *Region V* is symmetric to *Region IV*. Shortest path: $T2 * T1_P$; and
- (*T Region*) region defined by $T1_P$ and $T2_P$.

However, in [30] it has been shown that the previous taxonomy is only locally valid, *i.e.* near the desired configuration. As a consequence, Regions II, II', IV, V have to be further subdivided. Indeed, there exists a subspace of Regions IV and V, for which the shortest paths are $SL - T2 * T1_P$ or $SL - T1 * T2_P$ (see Regions VI and VII in Figure 18.8(b)), and a subspace of of Regions II, II' where the shortest paths are of kind $T1 * T2 - SL$ or $T2 * T1 - SL$.

18.4.1 IBVS and Optimal Paths

As the optimal paths are described and a taxonomy of the motion plane is derived, a visual-based controller able to track the optimal trajectories is needed in order to close the loop. Even though PBVS solutions can be adopted ([20]), the proposed IBVS solution avoids localization algorithms and then inherits intrinsic higher robustness with respect to PBVS approaches ([5]).

The IBVS scheme here proposed is feasible if: 1) optimal paths can be computed using only visual information and 2) once the vehicle optimal 3D words are "translated" to the image space. For the former point, it can be shown that only the orientation Ω between the initial position Q and the desired one P is needed ([30]). Since Ω is defined as $\int_0^\tau \omega(t)dt = \Omega$ along the optimal path, it can be estimated using epipolar geometry and the fundamental matrix F if at least eight feature points (in a non singular configuration) are given ([18]). Estimation robustness can be achieved if even more features are used.

18.4.1.1 Trajectories on the Image Plane

From the definition of the optimal paths in Section 18.4, it follows that only three specific kind of robot maneuvers needs to be translated: pure rotations, *i.e.*, $v = 0$, pure translations, *i.e.*, $\omega = 0$, and logarithm spirals. In what follows we assume that ${}^I p_i = [{}^I x_i, {}^I y_i]^T$, ${}^I p_c = [{}^I x_c, {}^I y_c]^T$ and ${}^I p_d = [{}^I x_d, {}^I y_d]^T$ are the feature initial, current and desired positions respectively.

Pure rotation: by plugging into the image Jacobian (18.4), the condition $v = 0$ and $\omega = \bar{\omega} =$ constant and solving for the ${}^I y_c$ feature coordinate, yields to

$$ {}^I y_c = \frac{{}^I y_i \cos\left(\arctan\left(\frac{{}^I x_i}{\alpha_x}\right)\right)}{\cos\left(\arctan\left(\frac{{}^I x_c}{\alpha_x}\right)\right)}, \tag{18.9} $$

that is the equation of a conic, *i.e.*, the intersection between the image plane and the cone with vertex in the camera center (optical center) and base circumference passing through the 3D feature position. On the other hand, solving for ${}^I x_c$, one gets

$$\theta - \theta_0 = \arctan\left(\frac{{}^I x_d}{\alpha_x}\right) - \arctan\left(\frac{{}^I x_i}{\alpha_x}\right), \tag{18.10}$$

where θ_0 is the initial value of θ, the robot orientation. Therefore, the angle variation can be computed for pure rotations by means of (18.10).
Pure translation: by plugging into the image Jacobian (18.4), the condition $\omega = 0$ and $v = \bar{v} =$ constant and solving for the ${}^I y_c$ feature coordinate, yields to

$${}^I y_c = \frac{{}^I y_i}{{}^I x_i} {}^I x_c, \tag{18.11}$$

which describes a straight line passing through the initial position of the feature and the principal point.
Logarithmic spiral: consider the condition ${}^I \dot{x}_c = 0$, *i.e.*, ${}^I x_c = {}^I \bar{x}_c =$ constant. Using (18.4), the condition reflects in

$$\omega = -\frac{\alpha_x}{\alpha_y \, {}^w y} \frac{{}^I \bar{x}^2 + \alpha_x^2}{{}^I \bar{x} \, {}^I y} v. \tag{18.12}$$

Since ${}^I x_c$ is constant, if follows that the angle γ in (18.8) between the forward direction of the vehicle and the feature direction is constant. Since this is the condition verified by a logarithmic spiral path, the condition (18.12) identifies such a path.

18.4.1.2 From Image Paths to Servoing

Once the alphabet of feature sub-paths has been defined, the rules to choose and construct the right sequence of maneuvers (the word) that correspond to the shortest path must be defined directly on the image plane. To this end, path of increasing complexity, *i.e.*, increasing number of symbols, are computed. If a path is feasible, *i.e.*, the feature path remains constrained inside the image view, than it is chosen. Otherwise, it is assumed that the robot state pertains to a different region in space and the algorithm is reiterated. The path solutions derived in Section 18.4.1.1 are computed numerically using the Levenberg–Marquardt algorithm ([30]).

The optimal feature path for *Region I* or *Region I'* is shown in Figure 18.9 and is composed of: a piece of a conic passing through the initial position of the feature (rotation); a piece of straight line passing through the principal point (translation); and a piece of conic passing through the final position of the feature (rotation). Given that Ω is known (or estimated), then the path can be computed solving the following equation

$$\theta_i + \theta_d = \Omega = \left[\arctan\left(\frac{{}^I x_1}{\alpha_x}\right) - \arctan\left(\frac{{}^I x_i}{\alpha_x}\right)\right] + \left[\arctan\left(\frac{{}^I x_d}{\alpha_x}\right) - \arctan\left(\frac{{}^I x_2}{\alpha_x}\right)\right],$$

where θ_i and θ_d are the orientation of the vehicle in the initial and final positions, the variables ${}^I x_1$ or ${}^I x_2$ correspond to the intermediate (unknown) positions of the

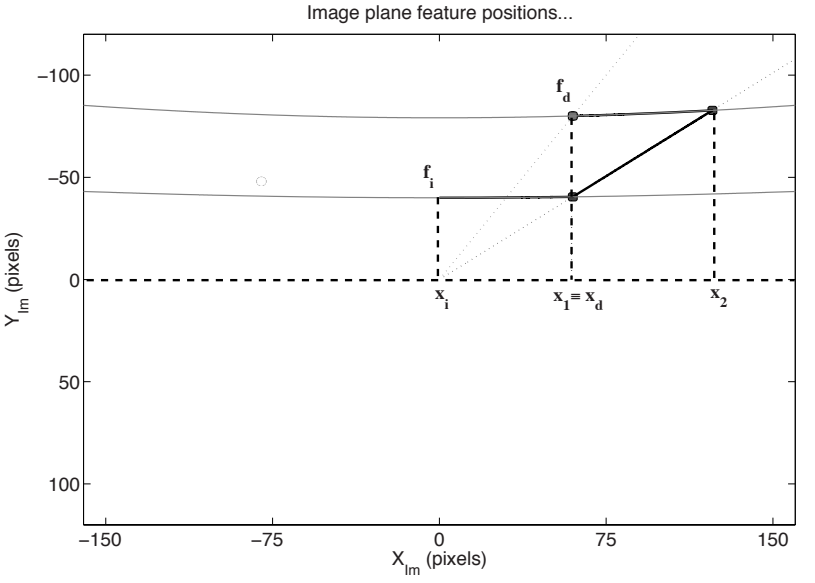


Fig. 18.9 Unconstrained optimal path for Region I in the image plane

feature along the path and are the initial and final feature position for the intermediate straight path.

The optimal feature path for the *Region III* is depicted in Figure 18.10 and it is determined solving the following equation

$$\Omega = \left[\arctan\left(\frac{{}^{I}x_1}{\alpha_x}\right) - \arctan\left(\frac{{}^{I}x_i}{\alpha_x}\right)\right] + \\ + \left[\frac{x_b}{\alpha_x}\ln\left(\frac{{}^{I}y_2}{{}^{I}y_3}\right)\right] + \left[\arctan\left(\frac{{}^{I}x_d}{\alpha_x}\right) - \arctan\left(\frac{x_b}{\alpha_x}\right)\right],$$

where the second addendum in the second member corresponds to the angle variation of the robot's orientation θ during the spiral path. x_b is the horizontal image bound.

Finally, the optimal feature paths that correspond to the regions *IV, V, VI* or *VII* are plotted in Figure 18.11. The solution in this case is given by

$$\Omega = \left[\arctan\left(\frac{{}^{I}x_1}{\alpha_x}\right) - \arctan\left(\frac{{}^{I}x_i}{\alpha_x}\right)\right] + \left[\frac{x_b}{\alpha_x}\ln\left(\frac{{}^{I}y_2}{{}^{I}y_3}\right)\right] + \\ + 2\left[\frac{x_b}{\alpha_x}\ln\left(\frac{{}^{I}y_3}{{}^{I}y_4}\right)\right] + \left[\arctan\left(\frac{{}^{I}x_d}{\alpha_x}\right) - \arctan\left(\frac{x_b}{\alpha_x}\right)\right].$$

Notice that, for each ${}^{I}y_4$, there exists a valid image path. To disambiguate, the choice of the optimal path in this last case can be done only by the use of its 3-D reconstruction (since both the path $T1-T2_P$ or $SL-T1-T2_P$ are feasible). It is

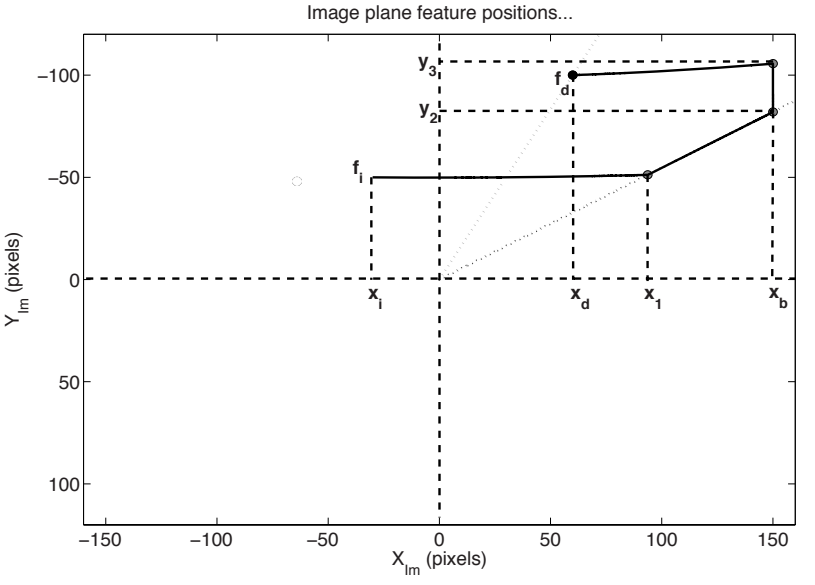


Fig. 18.10 Constrained optimal path for Region III in the image plane

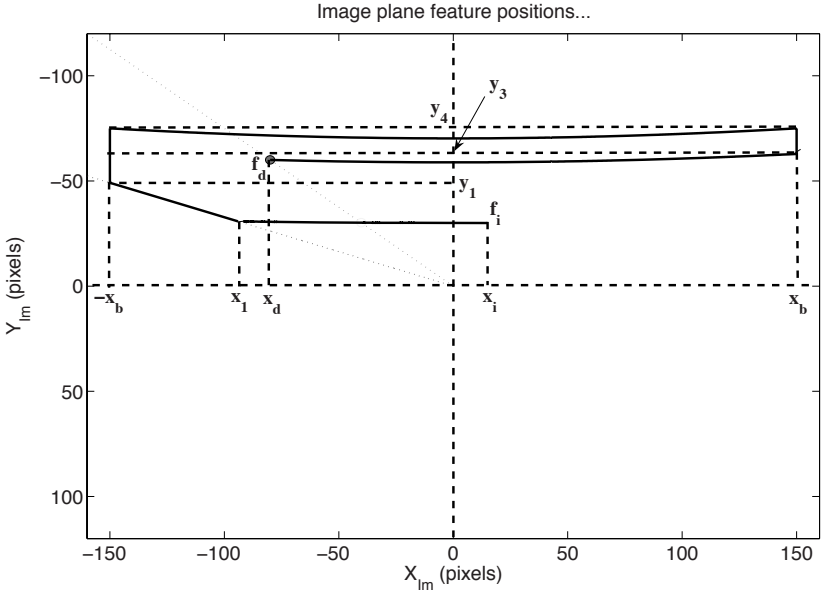


Fig. 18.11 One feature optimal path for Region VII in the image plane

worthwhile to note that such a reconstruction does not need to be exact, since a scaled one is sufficient [30].

Finally, the robot's proportional control laws $u = (v, \omega)$ are obtained using Lyapunov theory. For each image trajectory component a different Lyapunov function is chosen in order to minimize the feature tracking error.

18.4.2 Experimental Results

Experimental results for the IBVS and optimal path framework are now presented. The experimental setup comprises of a Quickcam Ultravision camera, whose resolution is 320x240 pixels, mounted over the front-part of a K-team Koala vehicle. The ERSP vision library ([17]) is used to perform SIFT recognition. The controller bandwidth is almost 7 Hz.

In the first experiment, the optimal path *SL* reported in Figure 18.9 is accomplished. The path is composed of the following maneuvers: a counter-clockwise rotation on the spot (the image conic ${}^I x_i \rightarrow {}^I x_1$), a forward motion (the straight line ${}^I x_1 \rightarrow {}^I x_2$), and a clockwise rotation on the spot (the image conic ${}^I x_2 \rightarrow {}^I x_d$). Figure 18.12 reports the related experimental result.

In the second experiment, the optimal path $SL - T1_P$ reported in Figure 18.10 is accomplished. The path is composed of the following sequence of maneuvers: a counter-clockwise rotation on the spot (the image conic ${}^I x_i \rightarrow {}^I x_1$), a forward motion (the straight line ${}^I x_1 \rightarrow {}^I x_b$), a logarithmic spiral (the straight line ${}^I y_2 \rightarrow {}^I y_3$) and a clockwise rotation on the spot (the image conic ${}^I x_b \rightarrow {}^I x_d$). Figure 18.13 reports the related experimental result.

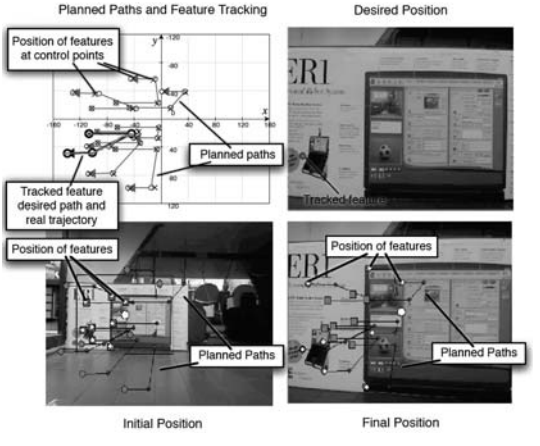


Fig. 18.12 Experiment 1. Planned paths for all features and the trajectory of the tracked feature (up left). Initial (bottom left), final (bottom right) and desired (up right) images taken from the vehicle. The planned paths and also the actual position of the features are plotted over the initial and final images

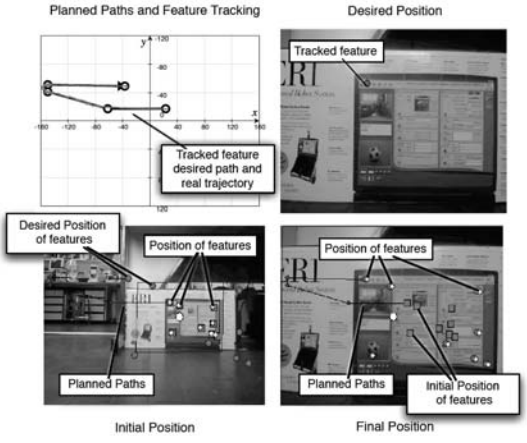


Fig. 18.13 Experiment 2. Planned paths and trajectory of the tracked feature (up left). Initial (bottom left), final (bottom right) and desired (up right) images taken from the vehicle. The initial and desired positions of the features are plotted over the initial and final images taken from the vehicle. The actual positions of the features are also shown in these images

The method accomplished the task of driving the tracked feature through the planned path. It can be observed from the figures that the final positions of the features are almost coincident with the desired positions. The final feature position error is due to the tracking noise and to the estimation of the angle Δ (that determines the feature trajectories). Hence, the final position of the robot is close to the desired one and the error is mostly due to a translation along the Z_w axis. The observed errors can be explained by image quantization deviations, presence of noisy data, camera specifications, low number of features used to compute the planned trajectory, estimation errors related to the SIFT recognition system, erroneous camera calibration parameters and, finally, the fact that the robot control was based on the tracking of just one feature in the image.

It is worthwhile to note that the previously presented PBVS VSLAM for servoing can be compared to the IBVS controller with a VSLAM architecture. Indeed, due to the very nature of the proposed solution, pure appearance-based visual maps can be adopted, without considering any 3D spatial information (*i.e.*, metric maps). Due to space limits, the interested reader is referred to [15].

18.5 Conclusions

The problem of visually guide unicycle-like vehicles towards desired position in space has been analyzed in this chapter. Solutions here discussed ranges from PBVS to IBVS approaches applied "in the large", which differs in which image data are treated in the control procedure.

For position based, we have proposed a visual servoing scheme for a nonholonomic vehicle in unknown indoor environments. The proposed approach is divided

into two phases: a servoing process and a map building for servoing process. The servoing hybrid control is based on a PBVS scheme that is also presented. Map construction complies with the need of overcoming the limits of the servoing scheme in a large environment. The work could be regarded as an attempt to connect control techniques (action) and sensorial data interpretation (perception).

Furthermore, a method that associates space regions to optimal vehicle trajectories, combined with a limited FOV camera, is presented. Then, optimal trajectory words are translated from 3D space to image space, in order to adopt an IBVS controller to track image feature trajectories. Notice that, the mobile robot autonomous capability, *i.e.*, the ability to move in a large environment, is increased using appearance-based visual maps, in which metric data are not of interest.

Experiments on real nonholonomic robot platforms are reported for both approaches. The robot successfully reached the desired position while keeping the tracked feature inside the FOV. Moreover, even if the initial and the desired images do not have a common set of features, the servoing is still feasible using image data stored in the maps.

References

[1] Benhimane, S., Malis, E.: A new approach to vision–based robot control with omni–directional cameras. In: Proc. IEEE Int. Conf. on Robotics and Automation, Orlando, Florida, USA, pp. 526–531 (2006)

[2] Bhattacharya, S., Murrieta-Cid, R., Hutchinson, S.: Optimal paths for landmark–based navigation by differential–drive vehicles with field–of–view constraints. IEEE Transactions on Robotics 23(1), 47–59 (2007)

[3] Brockett, R.: Asymptotic stability and feedback stabilization. In: Brockett, M. (ed.) Differential Geometric Control Theory, pp. 181–191. Birkhauser, Boston (1983)

[4] Chaumette, F., Hutchinson, S.: Visual servo control, Part I: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)

[5] Chaumette, F., Hutchinson, S.: Visual servo control, Part II: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)

[6] Chen, J., Dawson, D., Dixon, W., Chitrakaran, V.: Navigation function-based visual servo control. Automatica 43(7), 1165–1177 (2007)

[7] Chesi, G., Hashimoto, K., Prattichizzo, D., Vicino, A.: A swiching control law for keeping features in the field of view in eye-in-hand visual servoing. In: Proc. IEEE Int. Conf. on Robotics and Automation, Taipei, Taiwan, pp. 3929–3934 (2003)

[8] Chesi, G., Hung, Y.: Global path-planning for constrained and optimal visual servoing. IEEE Transactions on Robotics 23(5), 1050–1060 (2007)

[9] Chesi, G., Hung, Y.: Visual servoing: a global path-planning approach. In: Proc. IEEE Intl. Conf. on Robotics and Automation, Roma, Italy, pp. 2086–2091 (2007)

[10] Chiuso, A., Favaro, P., Jin, H., Soatto, S.: Structure from motion casually integrated over time. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(4), 523–535 (2002)

[11] Collewet, C., Chaumette, F.: Positioning a camera with respect to planar objects of unknown shape by coupling 2-d visual servoing and 3-d estimations. IEEE Trans. on Robotics and Automation 18(3) (2002)

[12] Davison, A.J.: Real–time simultaneous localization and mapping with a single camera. In: Proc. IEEE Int. Conf. on Computer Vision, vol. 2, pp. 1403–1410 (2003)
[13] Deguchi, K.: Optimal motion control for image-based visual servoing by decoupling translation and rotation. In: IEEE/RJS Intl. Conf. on Intelligent Robots and Systems, Victoria, B.C., Canada, pp. 705–711 (1998)
[14] Fontanelli, D., Danesi, A., Belo, F., Salaris, P., Bicchi, A.: Visual Servoing in the Large. The International Journal of Robotics Research 28(6), 802–814 (2009)
[15] Fontanelli, D., Salaris, P., Belo, F., Bicchi, A.: Visual appearance mapping for optimal vision based servoing. In: Experimental Robotics. Springer Tracts in Advanced Robotics (STAR), vol. 54, pp. 353–362. Springer, Heidelberg (2009)
[16] Gans, N., Hutchinson, S.: Stable visual servoing through hybrid switched system control. IEEE Transactions on Robotics 23(3), 530–540 (2007)
[17] Goncalves, L., DiBernardo, E., Benson, D., Svedman, M., Ostrowski, J., Karlsson, N., Pirjanian, P.: A visual front–end for simultaneous localization and mapping. In: Proc. IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, pp. 44–49 (2005)
[18] Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2003)
[19] Karlsson, N., DiBernardo, E., Ostrowski, J., Goncalves, L., Pirjanian, P., Munich, M.E.: The vSLAM algorithm for robust localization and mapping. In: Proc. IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, pp. 24–29 (2005)
[20] López-Nicolás, G., Bhattacharya, S., Guerrero, J., Sagüés, C., Hutchinson, S.: Switched homography-based visual control of differential drive vehicles with field-of-view constraints. In: Proc. IEEE Int. Conf. on Robotics and Automation, Rome, pp. 4238–4244 (2007)
[21] Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. Jour. of Computer Vision 20, 91–110 (2003)
[22] Lu, L., Dai, X., Hager, G.D.: Efficient particle filtering using RANSAC with application to 3D face tracking. Image and Vision Computing 24(6), 581–592 (2006)
[23] Mariottini, G.L., Oriolo, G., Prattichizzo, D.: Image–based visual servoing for nonholonomic mobile robots using epipolar geometry. IEEE Transactions on Robotics 23(1), 87–100 (2007)
[24] Mezouar, Y., Chaumette, F.: Path planning for robust image–based control. IEEE Transactions on Robotics and Automation 18(4), 534–549 (2002)
[25] Mezouar, Y., Remazeilles, A., Gros, P., Chaumette, F.: Images interpolation for image–based control under large displacement. In: Proc. IEEE Int. Conf. on Robotics and Automation, Washington, DC, pp. 3787–3794 (2002)
[26] Murrieri, P., Fontanelli, D., Bicchi, A.: A hybrid-control approach to the parking problem of a wheeled vehicle using limited view-angle visual feedback. Int. Jour. of Robotics Research 23(4-5), 437–448 (2004)
[27] Rekleitis, I., Sim, R., Dudek, G., Milios, E.: Collaborative exploration for the construction of visual maps. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1269–1274 (2001)
[28] Remazeilles, A., Chaumette, F., Gros, P.: Robot motion control from a visual memory. In: Proc. IEEE Int. Conf. on Robotics and Automation, vol. 5, pp. 4695–4700 (2004)
[29] Royer, E., Bom, J., Dhome, M., Thuilot, B., Lhuillier, M., Marmoiton, F.: Outdoor autonomous navigation using monocular vision. In: Proc. of the IEEE/RSJ Conference on Intelligent Robots and Systems, pp. 1253–1258 (2005)

[30] Salaris, P., Belo, F., Fontanelli, D., Greco, L., Bicchi, A.: Optimal paths in a constrained image plane for purely image–based parking. In: Proc. IEEE Int. Symp. Intelligent Robots and Systems, Nice, France, pp. 1673–1680 (2008)
[31] Sanderson, A.C., Weiss, L.E.: Image–based visual servo control using relational graph error signals. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1074–1077 (1980)
[32] Vedaldi, A., Jin, H., Favaro, P., Soatto, S.: KALMANSAC: Robust filtering by consensus. In: Proc. of the Int. Conf. on Computer Vision (ICCV), vol. 1, pp. 633–640 (2005)

Chapter 19
Unmanned Helicopter Control via Visual Servoing with Occlusion Handling

Yasushi Iwatani, Kei Watanabe, and Koichi Hashimoto

Abstract. This chapter presents an automatic take-off and landing system for an unmanned helicopter using visual servo control. Cameras are placed on the ground, and a static-eye configuration is set up. A vision-based control method with occlusion handling is implemented. The reference for the altitude in take-offs and landings is specialized to operate in ground effect. An experimental result demonstrates that the helicopter is controlled accurately.

19.1 Introduction

Computer vision plays an important role for automatic control of unmanned aerial vehicles and micro aerial vehicles [3, 4, 9]. In particular, cameras are useful to measure the position of a vehicle relative to the world reference frame. Vision-based control is also powerful in take-offs and landings, since vision provides the precise position relative to a given landing pad.

This chapter mainly focuses on automatic take-off and landing control of an unmanned helicopter using vision sensors. Restricted to take-offs and landings, there are two possibilities for camera configurations. In a hand-eye configuration, cameras are mounted on the helicopter. In a static-eye configuration, cameras are placed on the ground. Vision-based control in static-eye configurations is more robust to calibration errors than hand-eye configurations [1]. Thus a static-eye configuration is set up in this paper, while previous vision-based landing control systems for unmanned helicopters have hand-eye configurations [5, 6, 7].

In this chapter, visual servo control with occlusion handling proposed in [2] is implemented to avoid the occlusion problem caused by the rotor motion. The implemented method first chooses a set of correctly extracted image features in the selection step. It then estimates all the image features from the correctly extracted

Yasushi Iwatani, Kei Watanabe, and Koichi Hashimoto
Tohoku University, Aoba-ku Aramaki Aza Aoba 6-6-01, Sendai, Japan
e-mail: {iwatani,watanabe,koichi}@ic.is.tohoku.ac.jp

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 361–374.
springerlink.com

image features. The estimation procedure enables us to track image features even when non-predicted occlusion occurs. The implemented method integrates the image feature selection, the image feature estimation and reference tracking. The reference for the altitude in take-offs and landings is specialized to operate in ground effect. An experimental result demonstrates that the accuracy is well enough for take-offs and landings.

19.2 Experimental Setup

The system discussed in this chapter consists of an unmanned helicopter, four stationary cameras, three computers, a digital analog (DA) converter and a transmitter as illustrated in Figure 19.1. The helicopter does not have any sensors which measure the position or posture. It has four small black balls, and they are attached to rods connected to the bottom of the helicopter. The black balls are indexed from 1 to 4. Their positions in the image planes are used as image features. The four cameras are placed on the ground. Two of them look downward and the others look upward. The downward looking cameras are mainly used for take-off and landing control, while the upward looking cameras are used for hovering control. The cameras are indexed from 1 to 4. Each two of four cameras are connected to a computer.

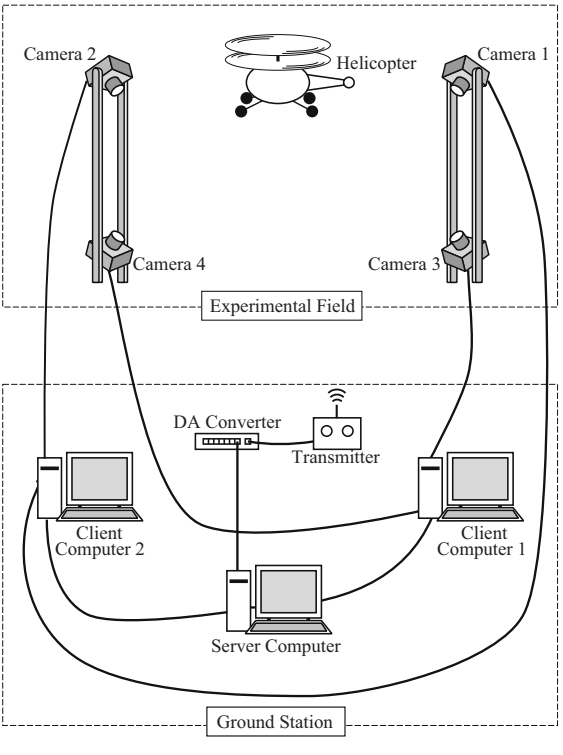


Fig. 19.1 System configuration. The system consists of a micro helicopter, four cameras, three computers, a DA converter and a transmitter

Fig. 19.2 X.R.B with four black balls. The positions of the black balls in the image planes are used as image features

The two computers connected to cameras are called client computers. The computer connected to the DA converter is called server computer.

Client computers extract image features and update the vectors of the image features in the image planes every 8.5 ms. This follows from the use of fast IEEE 1394 cameras, Dragonfly Express[1,2]. The vectors of the extracted image features are transferred to server computer.

Server computer determines a set of correctly extracted image features, or equivalently visible image features, at each time step by using the algorithm proposed in [2]. It also derives a required control signals in a manner presented in Section 19.7. The control signals are supplied from server computer to the helicopter through the DA converter and the transmitter. The clocks of the computers are not synchronized with each other. Server computer uses the latest data transferred from client computers. It takes 3 ms to update the control signals.

The unmanned helicopter used in experiments is X. R. B SR SKY ROBO Shuttle (see Figure 19.2). It has a coaxial rotor configuration. The two rotors share the same axis, and they rotate in opposite directions. The tail is a dummy. A stabilizer is installed on the upper rotor head. It mechanically keeps the posture horizontal.

Table 19.1 summarizes specifications of the system.

Table 19.1 Specifications of the system

Length of the helicopter	0.40 m
Height of the helicopter	0.21 m
Rotor length of the helicopter	0.34 m
Weight of the helicopter	0.21 kg
Focal length of the lens	0.004 m
Camera resolution	640×480 pixels
Pixel size	$7.4\ \mu\text{m} \times 7.4\ \mu\text{m}$

[1] Dragonfly Express is a trademark of Point Grey Research Inc.

[2] Point Grey Research™ and Dragonfly Express™ are trademarks owned by Point Grey Research, Inc. www.ptgrey.com

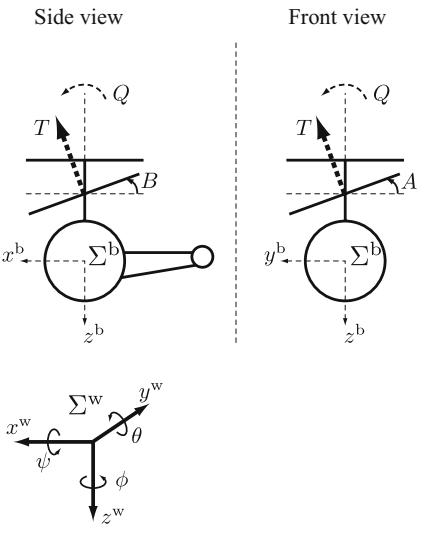


Fig. 19.3 Coordinate frames. The world reference frame and the helicopter coordinate frame are denoted by Σ^w and Σ^b, respectively. The origin of the z^w axis is set on the ground

19.3 Coordinate Frames

Let Σ^w be the world reference frame. The z^w axis is directed vertically downward. The origin of the z^w axis is set on the ground. A coordinate frame Σ^b is attached to the helicopter body as illustrated in Figure 19.3. The (x^b, y^b, z^b) positions of the four black balls in the frame Σ^b are given by

$$\begin{bmatrix} 0.1 \\ 0.1 \\ 0.04 \end{bmatrix}, \begin{bmatrix} -0.1 \\ 0.1 \\ 0.04 \end{bmatrix}, \begin{bmatrix} 0.1 \\ -0.1 \\ 0.04 \end{bmatrix}, \begin{bmatrix} -0.1 \\ -0.1 \\ 0.04 \end{bmatrix}. \tag{19.1}$$

Let $\boldsymbol{\xi}_{4(i-1)+j}$ denote the position of ball j in the image plane of camera i. The helicopter position relative to the world reference frame Σ^w is denoted by (x, y, z). The roll, pitch and yaw angles are denoted by ψ, θ and ϕ, respectively. The vector of generalized coordinates of the helicopter is defined by

$$\boldsymbol{q} = \begin{bmatrix} x & y & z & \phi \end{bmatrix}^\top. \tag{19.2}$$

Recall that the helicopter has a horizontal-keeping stabilizer. Both the angles θ and ψ converge to zero fast enough even when the body is inclined. Thus we may suppose in practice that

$$\theta(k) = 0 \ \text{ and } \ \psi(k) = 0, \ \forall\, k \geq 0. \tag{19.3}$$

The following four variables are individually controlled by control signals supplied to the transmitter (see also Figure 19.3):

B : elevator, pitch angle of the lower rotor.

A : aileron, roll angle of the lower rotor.

T : throttle, resultant force of the two rotor thrusts.

Q : rudder, difference of the two torques generated by the two rotors.

The corresponding control signals are denoted by u_1, u_2, u_3 and u_4. Note that x, y, z and ϕ are controlled by applying u_1, u_2, u_3 and u_4, respectively.

19.4 Image Jacobian Matrices

This section defines some image Jacobian matrices to explain our visual servoing technique in subsequent sections.

We have sixteen image features, since there exist four cameras in the flight field and the positions of the four black balls attached to the helicopter are used as image features. We set

$$\boldsymbol{\xi} = \left[\boldsymbol{\xi}_1^\top \ \boldsymbol{\xi}_2^\top \ \cdots \ \boldsymbol{\xi}_{16}^\top\right]^\top \tag{19.4}$$

where recall that $\boldsymbol{\xi}_{4(i-1)+j} \in \mathbb{R}^2$ denote the position of ball j in the image plane of camera i. In addition, we define

$$\boldsymbol{\xi}_{\mathbb{I}} = \left[\boldsymbol{\xi}_{\sigma_1}^\top \ \boldsymbol{\xi}_{\sigma_2}^\top \ \cdots \ \boldsymbol{\xi}_{\sigma_{|\mathbb{I}|}}^\top\right]^\top \tag{19.5}$$

where $\mathbb{I} \subset \{1,2\ldots,16\}$, $|\mathbb{I}|$ is the size of the set $\mathbb{I}$, $\sigma_j \in \mathbb{I}$ $(j = 1,2,\ldots,|\mathbb{I}|)$ and

$$\sigma_1 < \sigma_2 < \ldots < \sigma_{|\mathbb{I}|}. \tag{19.6}$$

As we shall see later, $\mathbb{I}$ implies a set of correctly extracted image features. The vector $\boldsymbol{\xi}_{\mathbb{I}}$ is used to generate a control input, when $\boldsymbol{\xi}_i$ with $i \in \mathbb{I}$ can be extracted correctly and when $\boldsymbol{\xi}_i$ with $i \notin \mathbb{I}$ can not be extracted due to occlusion. Details will be discussed in Section 19.6.

Using a nonlinear function $\boldsymbol{\alpha}_i$ of $\boldsymbol{q}$, we can write

$$\boldsymbol{\xi}_i = \boldsymbol{\alpha}_i(\boldsymbol{q}) \tag{19.7}$$

when occlusion does not occur (see for instance [8]). We define

$$\boldsymbol{\alpha}(\boldsymbol{q}) = \left[\boldsymbol{\alpha}_1^\top(\boldsymbol{q}) \ \boldsymbol{\alpha}_2^\top(\boldsymbol{q}) \ \ldots \ \boldsymbol{\alpha}_{16}^\top(\boldsymbol{q})\right]^\top \tag{19.8}$$

and we have

$$\boldsymbol{\xi} = \boldsymbol{\alpha}(\boldsymbol{q}). \tag{19.9}$$

Differentiating the above equation yields

$$\dot{\boldsymbol{\xi}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{19.10}$$

where

$$\boldsymbol{J}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\alpha}(\boldsymbol{q})}{\partial \boldsymbol{q}}. \tag{19.11}$$

The matrix function $\boldsymbol{J}$ defined by (19.11) is called the image Jacobian for $\boldsymbol{\xi}$. Similarly, we define the image Jacobian for $\boldsymbol{\xi}_{\mathbb{I}}$ by

$$\boldsymbol{J}_{\mathbb{I}}(\boldsymbol{q}) = \frac{\partial \boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q})}{\partial \boldsymbol{q}} \tag{19.12}$$

where

$$\boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q}) = \left[\boldsymbol{\alpha}_{\sigma_1}^{\top}(\boldsymbol{q})\ \boldsymbol{\alpha}_{\sigma_2}^{\top}(\boldsymbol{q})\ \cdots\ \boldsymbol{\alpha}_{\sigma_{|\mathbb{I}|}}^{\top}(\boldsymbol{q})\right]^{\top}. \tag{19.13}$$

19.5 Image Feature Estimation and Selection

This section discusses image feature estimation and selection. Section 19.5.1 provides a method to obtain an estimate of $\boldsymbol{\xi}$ from $\boldsymbol{\xi}_{\mathbb{I}}$. Section 19.5.2 presents a selection method to obtain a set of correctly extracted image features.

19.5.1 Image Feature Estimation

We first describe a reconstruction algorithm of generalized coordinates $\boldsymbol{q}$ by using the image Jacobian matrices. Let two vectors $\boldsymbol{\xi}^{\mathrm{o}}$ and $\boldsymbol{q}^{\mathrm{o}}$ which satisfy

$$\boldsymbol{\xi}^{\mathrm{o}} = \boldsymbol{\alpha}(\boldsymbol{q}^{\mathrm{o}}) \tag{19.14}$$

be given. A linear approximation of (19.9) near $\boldsymbol{q}^{\mathrm{o}}$ is given by

$$\boldsymbol{J}(\boldsymbol{q}^{\mathrm{o}})(\tilde{\boldsymbol{q}} - \boldsymbol{q}^{\mathrm{o}}) = \boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{o}} \tag{19.15}$$

or

$$\tilde{\boldsymbol{q}} = \boldsymbol{J}^{+}(\boldsymbol{q}^{\mathrm{o}})(\boldsymbol{\xi} - \boldsymbol{\xi}^{\mathrm{o}}) + \boldsymbol{q}^{\mathrm{o}} \tag{19.16}$$

where $\tilde{\boldsymbol{q}}$ is an approximate value of $\boldsymbol{q}$, and $\boldsymbol{J}^{+}(\boldsymbol{q}^{\mathrm{o}})$ the Moore–Penrose inverse of $\boldsymbol{J}$ at $\boldsymbol{q}^{\mathrm{o}}$. An approximate value of $\boldsymbol{q}$ is also obtained by

$$\boldsymbol{J}_{\mathbb{I}}(\boldsymbol{q}^{\mathrm{o}})(\tilde{\boldsymbol{q}} - \boldsymbol{q}^{\mathrm{o}}) = \boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}} \tag{19.17}$$

or

$$\tilde{\boldsymbol{q}} = \boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{q}^{\mathrm{o}})(\boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}) + \boldsymbol{q}^{\mathrm{o}} \tag{19.18}$$

where $\boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}$ is defined by replacing $\boldsymbol{\xi}$ with $\boldsymbol{\xi}^{\mathrm{o}}$ in (19.5). To simplify notation, we write $\boldsymbol{J}$, $\boldsymbol{J}^{+}$, $\boldsymbol{J}_{\mathbb{I}}$ and $\boldsymbol{J}_{\mathbb{I}}^{+}$ instead of $\boldsymbol{J}(\boldsymbol{q}^{\mathrm{o}})$, $\boldsymbol{J}^{+}(\boldsymbol{q}^{\mathrm{o}})$, $\boldsymbol{J}_{\mathbb{I}}(\boldsymbol{q}^{\mathrm{o}})$ and $\boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{q}^{\mathrm{o}})$, respectively, in this section.

We next obtain an estimate of $\boldsymbol{\xi}$ from $\boldsymbol{\xi}_{\mathbb{I}}$. Substituting (19.18) into (19.15), we have

$$\hat{\boldsymbol{\xi}} = \boldsymbol{J}\boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}) + \boldsymbol{\xi}^{\mathrm{o}}. \tag{19.19}$$

The vector $\hat{\boldsymbol{\xi}}$ implies an estimate of all the image features from correctly extracted image features, if $\mathbb{I}$ is a set of correctly extracted image features. We here define

$$\left[\hat{\boldsymbol{\xi}}_1^{\top}\ \hat{\boldsymbol{\xi}}_2^{\top}\ \cdots\ \hat{\boldsymbol{\xi}}_{16}^{\top}\right]^{\top} = \hat{\boldsymbol{\xi}}, \tag{19.20}$$

$$\hat{\boldsymbol{\xi}}_i \in \mathbb{R}^2, \quad i \in \{1,2,\ldots,16\}. \tag{19.21}$$

19.5.2 Image Feature Selection

Similar to (19.19), we can estimate $\boldsymbol{\xi}_{\mathbb{I}}$ from itself by

$$\tilde{\boldsymbol{\xi}}_{\mathbb{I}} = \boldsymbol{J}_{\mathbb{I}}\boldsymbol{J}_{\mathbb{I}}^{+}(\boldsymbol{\xi}_{\mathbb{I}} - \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}) + \boldsymbol{\xi}_{\mathbb{I}}^{\mathrm{o}}. \tag{19.22}$$

We here set

$$\left[\tilde{\boldsymbol{\xi}}_{\sigma_1}^{\top}\ \tilde{\boldsymbol{\xi}}_{\sigma_2}^{\top}\ \cdots\ \tilde{\boldsymbol{\xi}}_{\sigma_{|\mathbb{I}|}}^{\top}\right]^{\top} = \tilde{\boldsymbol{\xi}}_{\mathbb{I}}, \tag{19.23}$$

$$\tilde{\boldsymbol{\xi}}_{\sigma_j} \in \mathbb{R}^2, \quad \sigma_j \in \mathbb{I}, \quad j \in \{1,2,\ldots,|\mathbb{I}|\}. \tag{19.24}$$

The vector $\tilde{\boldsymbol{\xi}}_{\sigma_j}$ implies an estimate of $\boldsymbol{\xi}_{\sigma_j}$ from $\boldsymbol{\xi}_{\mathbb{I}}$. If $\tilde{\boldsymbol{\xi}}_{\sigma_j}$ is far from $\boldsymbol{\xi}_{\sigma_j}$, then we may determine that the σ_j-th image feature is not extracted correctly due to occlusion. We therefore obtain a better candidate for a set of correctly extracted image features by the following procedure:

$$\ell = \arg\max_{\sigma_j \in \mathbb{I}} \|\boldsymbol{\xi}_{\sigma_j} - \tilde{\boldsymbol{\xi}}_{\sigma_j}\|, \tag{19.25}$$

$$\mathbb{I} \leftarrow \mathbb{I} - \ell. \tag{19.26}$$

If $\|\boldsymbol{\xi}_{\sigma_j} - \tilde{\boldsymbol{\xi}}_{\sigma_j}\|$ is within a given tolerance for all $\sigma_j \in \mathbb{I}$, then $\mathbb{I}$ implies a set of correctly extracted image features

19.6 Visual Servo Control with Occlusion Handling

This section shows a visual servo control algorithm implemented in our control system.

Several notations are defined to describe details of the implemented algorithm. The vector of the generalized coordinates at time k is denoted by $\boldsymbol{q}(k)$, and the vector

of all the image features at time k is written by $\boldsymbol{\xi}(k)$. We use $\boldsymbol{q}(k-1)$ as $\boldsymbol{q}^{\mathrm{o}}$ at time k. We thus use $\boldsymbol{\alpha}(\boldsymbol{q}(k-1))$ as $\boldsymbol{\xi}^{\mathrm{o}}$ at time k, which satisfies (19.14). The arguments of image Jacobian matrices and Moore–Penrose inverse of them are $\boldsymbol{q}(k-1)$, but the arguments are omitted to simplify notation.

The implemented algorithm consists of image feature extraction, image feature selection, update of the state, control input calculation and image feature estimation as illustrated in Figure 19.4, where the initial state $\boldsymbol{q}(0)$ is given. Let us now show details of the implemented algorithm:

1. *image feature extraction*: image features at time k are extracted by using an estimate of image features at time $k-1$. The center of a search area to extract $\boldsymbol{\xi}_i(k)$ is set at $\hat{\boldsymbol{\xi}}_i(k-1)$. The image feature $\boldsymbol{\xi}_i(k)$ is given by the center of gravity in the corresponding search area;
2. *image feature selection*: let the minimum size of $\mathbb{I}$, say m_{s}, and a tolerance ε (>0) be given. The following algorithm determines the set of visible image features $\mathbb{I}$:

```
𝕀 = {1,2,...,m}
for i = 1,2,...,m−1
  for j = i+1,2,...,m
    if the search areas of i and j are overlapped
      𝕀 ← 𝕀 − {i,j}
    end
  end
end
until |𝕀| ≥ m_s
  ξ̃_𝕀(k) = J_𝕀 J_𝕀^+ {ξ_𝕀(k) − α_𝕀(q(k−1))} + α_𝕀(q(k−1))
  if max_{i∈𝕀} ‖ξ_i(k) − ξ̃_i(k)‖ > ε
    ℓ = arg max_{i∈𝕀} ‖ξ_i(k) − ξ̃_i(k)‖
    𝕀 ← 𝕀 − {ℓ}
  else
    break
  end
end
```

3. *update of the state*: the state $\boldsymbol{q}(k)$ is given by

$$\boldsymbol{q}(k) = \boldsymbol{J}_{\mathbb{I}}^{+}\{\boldsymbol{\xi}_{\mathbb{I}}(k) - \boldsymbol{\alpha}_{\mathbb{I}}(\boldsymbol{q}(k-1))\} + \boldsymbol{q}(k-1); \tag{19.27}$$

4. *control input calculation*: a control input signal $\boldsymbol{u}(k)$ is determined by a set of proportional integral derivative (PID) controllers. Details are shown in the next section, since special reference signals are required for take-off and landing control;
5. *image feature estimation*: using (19.19), image features are estimated by

$$\hat{\boldsymbol{\xi}}(k) = \boldsymbol{J}\boldsymbol{J}_{\mathbb{I}}^{+}\{\boldsymbol{\xi}_{\mathbb{I}}(k) - \boldsymbol{\alpha}(\boldsymbol{q}(k-1))\} + \boldsymbol{\alpha}(\boldsymbol{q}(k-1)). \tag{19.28}$$

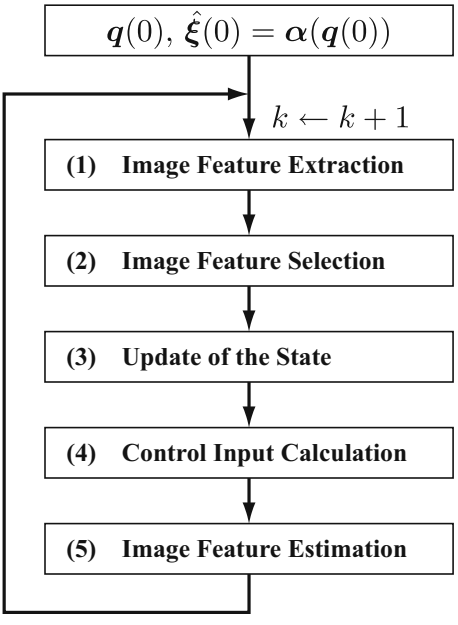


Fig. 19.4 A flow diagram outlining the proposed visual servo control system. The initial state $\boldsymbol{q}(0)$ is given

19.7 Controller Design

The goal here is to achieve

$$\boldsymbol{q}(k) \to \boldsymbol{q}^{\mathrm{ref}}(k) \tag{19.29}$$

where

$$\boldsymbol{q}^{\mathrm{ref}}(k) = \left[x^{\mathrm{ref}}(k)\ y^{\mathrm{ref}}(k)\ z^{\mathrm{ref}}(k)\ \phi^{\mathrm{ref}}(k)\right]^{\top} \tag{19.30}$$

is a given reference. We define

$$\left[\bar{q}_1(k)\ \bar{q}_2(k)\ \bar{q}_3(k)\ \bar{q}_4(k)\right]^{\top} = \boldsymbol{R}(\phi(k))(\boldsymbol{q}(k) - \boldsymbol{q}^{\mathrm{ref}}(k)) \tag{19.31}$$

where

$$\boldsymbol{R}(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 & 0 \\ -\sin\phi & \cos\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{19.32}$$

The rest of this section describes normal tracking control, take-off control and landing control methods.

For normal tracking control, the control signals are given by a set of PID controllers of the form

Table 19.2 PID gains

i	b_i	P_i	I_i	D_i
1	2.15	−3.80	−0.10	−2.0
2	2.67	3.80	0.10	2.0
3	1.85	3.40	1.00	1.8
4	2.39	1.50	0.80	0.1

$$u_i(k) = b_i + P_i\bar{q}_i(k) + I_i \sum_{\tau=0}^{k} \bar{q}_i(\tau) + D_i(\bar{q}_i(k) - \bar{q}_i(k-1)), \tag{19.33}$$

where b_i, P_i, I_i and D_i are constants for $i = 1, \ldots, 4$. The PID gains are heuristically tuned to the values in Table 19.2. The signs of the gains depend on the property of the transmitter. If the reference vector $\boldsymbol{q}^{\mathrm{ref}}$ is fixed over a period of time, then hovering motion is generated.

For take-off control, the following proportional derivative (PD) controllers are used:

$$u_i(k) = b_i + P_i\bar{q}_i(k) + D_i(\bar{q}_i(k) - \bar{q}_i(k-1)), \tag{19.34}$$

where b_i, P_i and D_i are set to the values in Table 19.2. In order to ensure a smooth take-off in ground effect, the reference $z^{\mathrm{ref}}(k)$ should be set underground at first. In our experiments, $z^{\mathrm{ref}}(k)$ varies from 0.1 m to −1 m smoothly, since the rotors stop when $z^{\mathrm{ref}}(k) = 0.1$ and ground effect vanishes at an altitude of −1 m. Recall that $z^{\mathrm{ref}}(k) > 0$ means that the reference is underground (see Figure 19.3).

For landing control, the PD controllers (19.34) are also used. The reference $z^{\mathrm{ref}}(k)$ increases until $z(k)$ is larger than a given threshold. In our experiments, $z^{\mathrm{ref}}(k)$ is set to 0.1 m, when $z(k) > -0.02$ m. Recall that the rotors stop when $z^{\mathrm{ref}}(k) = 0.1$.

19.8 Experimental Result

The reference of the generalized coordinates for hovering is set at $\begin{bmatrix} 0 & 0 & -1 & 0 \end{bmatrix}^\top$ with respect to the world reference frame Σ^{w}. The cameras are located as shown in Table 19.3 (The cameras are set as follows. Camera i is first set at the origin of Σ^{w}. The

Table 19.3 Camera configuration

camera index	x	y	z	ϕ	θ
1	−1.42	0.00	−0.97	0	$-7\pi/36$
2	0.00	−1.42	−0.97	$-\pi/2$	$-7\pi/36$
3	−1.43	0.00	−0.12	0	$\pi/9$
4	0.00	−1.43	−0.12	$-\pi/2$	$\pi/6$

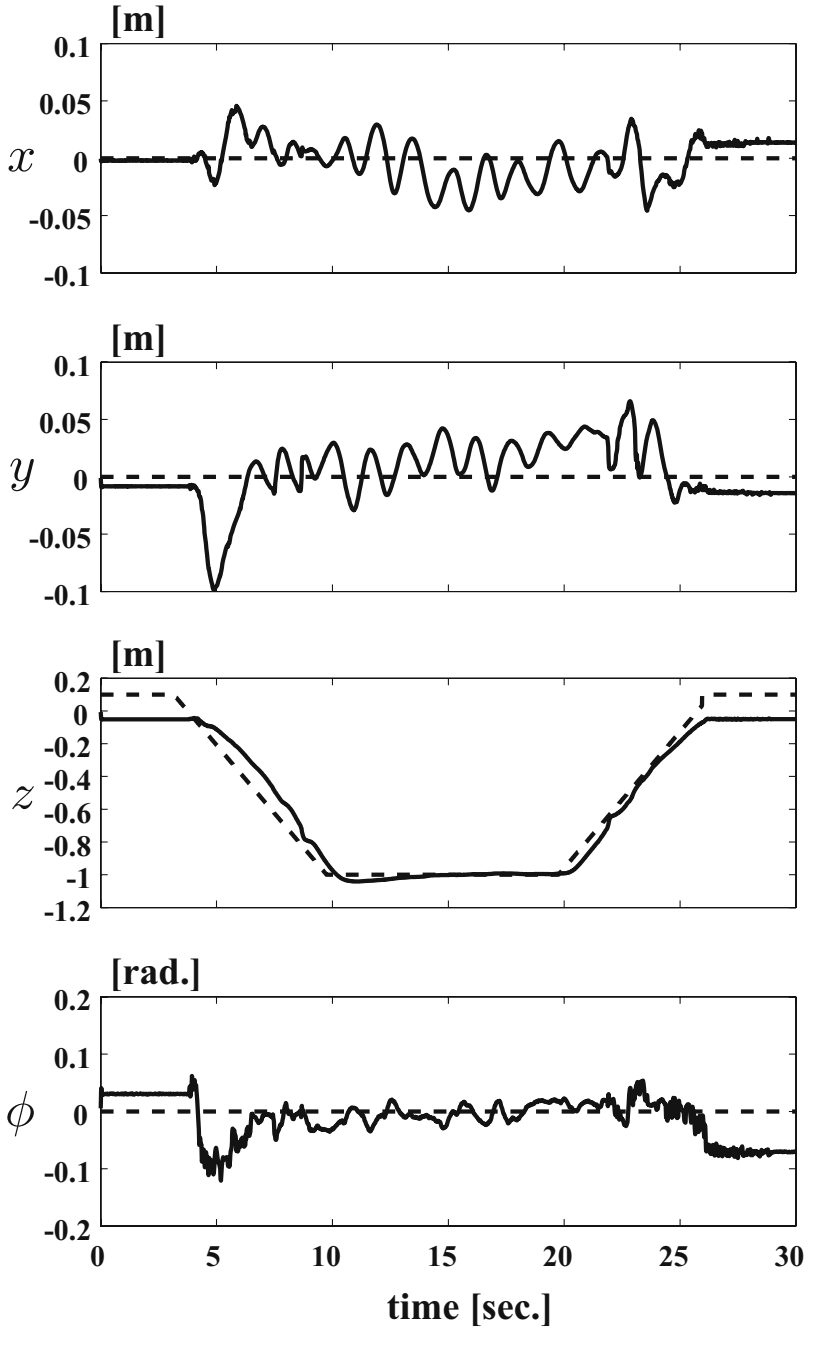


Fig. 19.5 Time profiles of generalized coordinates. The *solid lines* mean measurements, that is, $\boldsymbol{q}(k)$. The *dashed lines* denote references, that is, $\boldsymbol{q}^{\mathrm{ref}}(k)$

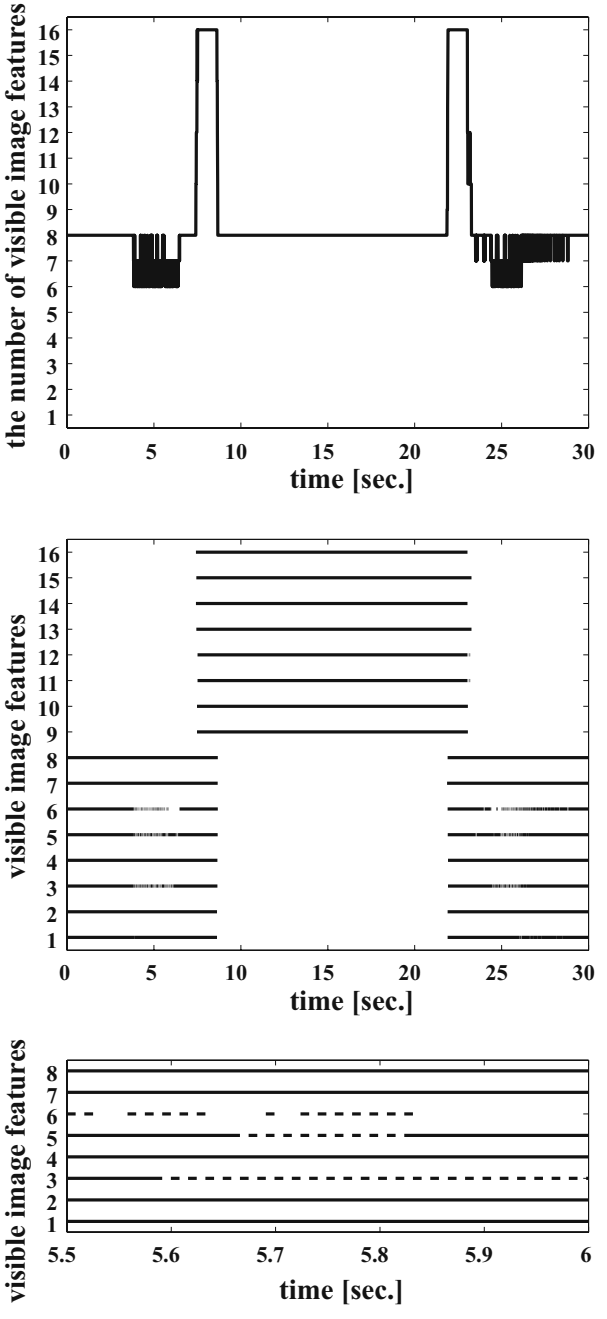


Fig. 19.6 Time profiles of visible image features. The line for i is drawn in the middle and bottom figures when $\boldsymbol{\xi}_i$ is correctly extracted or equivalently $\boldsymbol{\xi}_i$ is visible. The bottom is a closeup of the middle

optical axis of camera i lies along x^{w} axis. Camera i is then moved in accordance with Table 19.3).

Figures 19.5 and 19.6 show time profiles of generalized coordinates and visible image features. The control mode was switched from take-off control to normal tracking control at 10 s and from normal tracking control to landing control at 20 s. It is seen from Figure 19.5 that the helicopter was controlled well. In particular, the errors for x and y were within ± 0.1 m. Figure 19.6 shows that occlusion took place in the take-off and landing. The bottom of Figure 19.6 is a closeup of the middle. It is seem from the bottom figure that frequent changes occurred in the set of visible image features due to occlusion caused by the rotors. The experimental result also demonstrates the robustness of the vision servo control method with occlusion handling proposed in [2].

19.9 Conclusions

This chapter presented an automatic control method for take-offs, landings, and reference tracking of an unmanned helicopter using visual servoing with occlusion handling. The proposed control system provides high accuracy in take-offs, hovering and landings. In fact, an experimental result shows the helicopter is within 0.1 m from the reference, while the length of the helicopter is 0.4 m.

Several movies can be found at the webpage `http://www.ic.is.tohoku.ac.jp/E/research/helicopter/`. They show stability, convergence and robustness of the system in an easy-to-understand way, while the properties may not be seen easily from the figures in this chapter.

References

[1] Chesi, G., Hashimoto, K.: Configuration and robustness in visual servo. Journal of Robotics and Mechatronics 16(2), 178–185 (2004)

[2] Iwatani, Y., Watanebe, K., Hashimoto, K.: Image feature extraction with occlusion handling for visual servo control. In: IEEE International Conference on Robotics and Automation (2008)

[3] Ludington, B., Johnson, E., Vachtsevanos, G.: Augmenting UAV autonomy: vision-based navigation and target tracking for unmanned aerial vehicles. IEEE Robotics & Automation Magazine 13(3), 63–71 (2006)

[4] Merino, L., Wiklund, J., Caballero, F., Moe, A., Dios, J., Forssén, P., Nordberg, K., Ollero, A.: Vision-based multi-UAV position estimation: Localization based on blob features for exploration missions. IEEE Robotics & Automation Magazine 13(3), 53–62 (2006)

[5] Ruffier, F., Franceschini, N.: Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In: IEEE International Conference on Robotics and Automation (2004)

[6] Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Visually-guided landing of an unmanned aerial vehicle. IEEE Trans. on Robotics and Automation 19(3), 371–381 (2003)
[7] Shakernia, O., Sharp, C.S., Vidal, R., Shim, D.H., Ma, Y., Sastry, S.: Multiple view motion estimation and control for landing an unmanned aerial vehicle. In: IEEE International Conference on Robotics and Automation (2002)
[8] Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. Wiley, Chichester (2005)
[9] Watanabe, K., Yoshihata, Y., Iwatani, Y., Hashimoto, K.: Image-based visual PID control of a micro helicopter using a stationary camera. Advanced Robotics 22(2-3), 381–393 (2008)

Chapter 20
Visual Servoing via Nonlinear Predictive Control

Guillaume Allibert, Estelle Courtial, and François Chaumette

Abstract. In this chapter, image-based visual servoing is addressed via nonlinear model predictive control. The visual servoing task is formulated into a nonlinear optimization problem in the image plane. The proposed approach, named visual predictive control, can easily and explicitly take into account 2D and 3D constraints. Furthermore, the image prediction over a finite prediction horizon plays a crucial role for large displacements. This image prediction is obtained thanks to a model. The choice of this model is discussed. A nonlinear global model and a local model based on the interaction matrix are considered. Advantages and drawbacks of both models are pointed out. Finally, simulations obtained with a 6 degrees of freedom (DOF) free-flying camera highlight the capabilities and the efficiency of the proposed approach by a comparison with the classical image-based visual servoing.

20.1 Introduction

Visual servoing has lead to many fruitful researches over the last decades. In regard to the kind of feedback information considered, one can distinguish three main approaches: image-based visual servoing (IBVS) where the feedback is defined in the image plane, position-based visual servoing (PBVS) where the feedback is composed of 3D data such as the robotic system pose, and the 2 1/2 D visual servoing where the feedback combines both 2D and 3D data. Further details about the different approaches can be found in [5, 6, 15]. Here, we focus our interest on IBVS strategy. The IBVS task consists in determining the control input applied to the robotic system so that a set of visual features designed from image measurements

Guillaume Allibert and Estelle Courtial
Institut PRISME, Polytech'Orleans, 8 rue Leonard de Vinci, 45072 Orleans, France
e-mail: {guillaume.allibert,estelle.courtial}@univ-orleans.fr

François Chaumette
INRIA, Campus de Beaulieu, 35042 Rennes, France
e-mail: francois.chaumette@irisa.fr

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 375–393.
springerlink.com

reaches a desired static reference or follows a desired dynamic reference. Although IBVS approach is robust to modeling errors, several drawbacks can be mentioned when the visual features are not correctly chosen. Besides the classical problem of local minima and singularities in the interaction matrix [4], the constraint handling is a tricky problem in IBVS. For instance, the 2D constraint, also named visibility constraint, has to guarantee that the image measurements stay into the camera field of view. Of course, if the visibility of the target is no longer ensured then the control algorithm is interrupted. The 3D constraints such as workspace limits have to make sure that the robot achieves admissible motions in its workspace all along the task.

Among the numerous works which have investigated this critical issue, three points of view exist. The first one consists in designing adequate visual features. In [17] for instance, the authors have shown that the system behavior explicitly depends on the kind of features. Consequently, lines, spheres, circles, cylinders but also moments may be used and combined to obtain good decoupling and linearizing properties, implicitly ensuring the constraint handling. The control law is generally a decoupled exponential decreasing law. Another way to deal with constraint handling is to combine path-planning and trajectory tracking [7, 16, 19, 24]. When it is successful, this solution allows ensuring both an optimal trajectory of the camera in the Cartesian space and the visibility of the features. Path-planning via linear matrix inequality (LMI) optimization has recently been proposed in [7] to fulfill 2D and 3D constraints. In the third approach, the effort is done on the control law design. The visual features considered are generally basic, namely point-like features. Advanced control laws such as optimal control [14, 22], adaptive control [21], LMIs [9, 10] and predictive control [2, 3, 12, 13, 23] have been reported in the literature. In [12, 13], a predictive controller is used for motion compensation in target tracking applications. The prediction of the target motion is used to reject perturbation in order to cancel tracking errors. In [23], the predictive controller is used from ultrasound images for a medical application.

The strategy proposed in this chapter exploits nonlinear model predictive control for visual servoing tasks. The IBVS objective is formulated as solving on-line a nonlinear optimization problem expressed in the image plane [2, 3]. This strategy, named visual predictive control (VPC), offers two advantages. First, 2D and 3D constraints such as visibility constraints, mechanical constraints and workspace limits can be easily taken into account in the optimization problem. Secondly, the image prediction over a finite horizon plays a crucial role for difficult configurations. The image prediction is based on the knowledge of a model. It can be a nonlinear global model combining the robot model and the camera one. The image prediction can also be obtained thanks to a linear model using the interaction matrix. The choice of the model is addressed and discussed in the sequel. The interest of the image prediction is pointed out through many simulations describing difficult configurations for a free-flying perspective camera.

The chapter is organized as follows. In Section 20.2, the context of the study is stated and the principle of VPC is presented. The control structure and the mathematical formulation are addressed. Then, in Section 20.3, the choice of the image prediction model is discussed. In Section 20.4, numerous simulations on a 6 DOF

free-flying camera illustrate the comparison of the different approaches: classical IBVS, predictive control laws with local and global model. Difficult configurations such as large displacements to achieve are tested under constraints. Finally, conclusions are given in the last section.

20.2 Predictive Control for Constrained IBVS

The aim of visual servoing is to regulate to zero an error $e(t)$ between the current features $s(t)$ and the reference features $s^\star$. In IBVS, the features are expressed in the image. The relationship between the camera velocity $\tau(t)$ and the time variation of the visual features $\dot{s}(t)$ is given by the interaction matrix noted L_s. Thus, specifying a decoupled exponential decay law for the error $e(t)$, we obtain the control input to be applied to the camera:

$$\tau(t) = -\lambda \widehat{L_s}^{+} e(t) \text{ with } \lambda > 0, \tag{20.1}$$

where $\widehat{L_s}^{+}$ is the approximate pseudo-inverse matrix of L_s. The classical IBVS is very easy to implement but its weak points are the constraint handling and its possible bad behavior for large displacements to achieve as already mentioned in Section 20.1. The control objective of IBVS can also be formulated into an optimization problem. The goal is to minimize an image error and to take into account constraints. When a model of the system is available, control predictive strategies are well-adapted to deal with this kind of problem. The extension of predictive strategy to visual servoing tasks is detailed below.

20.2.1 Visual Predictive Control

All predictive strategies are based on four common points: a reference trajectory, a model of the dynamic process, a cost function and a solving optimization method. The keystone of the predictive approach is the model used to predict the process behavior over the future horizon. Its choice will impact on the tracking accuracy and on the computational time. In VPC case, the process considered is generally composed of the robotic system and the camera. For instance, the robotic system can be a nonholonomic mobile robot [2], a drone or a robot arm. The camera system can be a perspective or catadioptric camera [3] whatever its configuration with respect to the robot, that is on board or remote. The model used is then a global model describing the process. The model inputs are the control variables of the robotic system. The model outputs are the visual features. The model is used to predict the values of the features over a prediction horizon in regard to the control variables and to satisfy the constraint handling. Before discussing the choice of the model, we first introduce the control structure and then state the mathematical formulation of VPC.

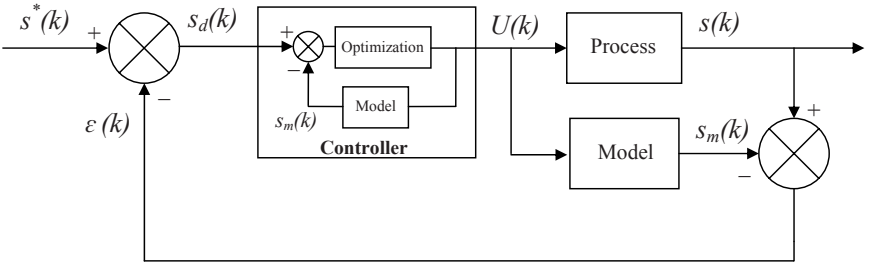


Fig. 20.1 IMC Structure

20.2.2 Internal Model Control Structure

The control structure considered is the well-known internal model control (IMC) structure [20] (see Figure 20.1). The process block contains the robotic system and the camera. The input U is the robotic control variable and the output s is the current value of the visual features. For IBVS, the reference s^* is expressed in the image plane, as the visual features, and can be static or dynamic. The error signal ε represents all modeling errors and disturbances between the current features and the values predicted from the model of the system:

$$\varepsilon(k) = s(k) - s_m(k). \tag{20.2}$$

The usual controller is replaced, in the predictive approach, by an optimization algorithm. The latter minimizes the difference between a desired trajectory s_d and the predicted model output s_m. Indeed, according to Figure 20.1, we can write (k is the current iteration)

$$\begin{aligned} s_d(k) &= s^*(k) - \varepsilon(k), \\ s_d(k) &= s^*(k) - (s(k) - s_m(k)), \\ s_d(k) - s_m(k) &= s^*(k) - s(k). \end{aligned} \tag{20.3}$$

Consequently, the tracking of the reference features s^* by the process output s is thus equivalent to the tracking of the desired features s_d by the model output s_m. The model predicts the behavior of the features over a finite prediction horizon N_p. The difference $s_d(k) - s_m(k)$ between the desired features and the predicted model features is used to define the cost function J to be minimized with respect to a control sequence $\tilde{U}$. Only the first component $U(k)$ of the optimal control sequence is really applied to the process. At the next sampling time, due to disturbances and model mismatches, the measurements are updated, the finite horizon moves one step forward and the procedure starts again.

20.2.3 Mathematical Formulation

The cost function J is defined as a quadratic function of the error to be minimized. Due to the IMC structure, the mathematical formulation of VPC strategy can be written in discrete-time as

$$\min_{\tilde{U}\in\mathbb{R}^{m\times N_p}} J(U) \tag{20.4}$$

with

$$J(U) = \sum_{j=k+1}^{k+Np} [s_d(j) - s_m(j)]^T \; Q(j) \; [s_d(j) - s_m(j)] \tag{20.5}$$

subject to

$$s_d(j) = s^*(j) - \varepsilon(j), \tag{20.6}$$

$$\begin{cases} x(j) = f(x(j-1), U(j-1)) \\ s_m(j) = h(x(j)). \end{cases} \tag{20.7}$$

The variables $x \in \mathbb{R}^n$, $U \in \mathbb{R}^m$ and $s_m \in \mathbb{R}^p$ are respectively the state, the input and the output of the model. We will see, in the next section, that the state can be differently chosen in regard to the prediction model used and in regard to the constraints to be handled. The first nonlinear equation of (20.7) describes the dynamics of the system where $x(j)$ represents the predicted state at time j, $\forall j \in [k+1; k+N_p]$. For $j = k+1$, the predicted state s_m is initialized with the system state s at time k which guarantees the feedback of the IMC structure. Moreover, in case of modeling errors and disturbances, a second feedback is ensured by the error signal $\varepsilon(j)$ which modifies the reference trajectory accordingly. The second equation of (20.7) is the output equation. To compute $s_d(j)$, $\forall j \in [k+1; k+N_p]$, we need to compute the error $\varepsilon(j)$ defined in (20.2). This error depends on $s_m(j)$ that is available but also on $s(j)$ that is unknown over the prediction horizon. Consequently, the error $\varepsilon(j)$ is assumed to be constant over the prediction horizon:

$$\varepsilon(j) = \varepsilon(k) = s(k) - s_m(k), \;\; \forall j \in [k+1; k+N_p]. \tag{20.8}$$

Finally, $\tilde{U} = \{U(k), U(k+1), ..., U(k+N_c), ..., U(k+N_p-1)\}$ is the optimal control sequence. From $U(k+N_c+1)$ to $U(k+N_p-1)$, the control input is constant and equal to $U(k+N_c)$ where N_c is the control horizon. The weighted matrix $Q(j)$ is a symmetric definite positive matrix.

One of the main advantages of VPC is the capability to explicitly handle constraints in the optimization problem. Three kinds of constraints are distinguished:

- constraints on the state of the robotic system. It can typically be a mechanical constraint such as workspace limit when the state represents the camera pose:

$$x_{\min} \leq x(k) \leq x_{\max}; \tag{20.9}$$

- 2D constraints also named visibility constraints to ensure that the visual features stay in the image plane or to represent forbidden areas in the image. The latter can be very useful to deal with obstacle avoidance or image occlusion:

$$s_{\min} \leq s_m(k) \leq s_{\max}; \tag{20.10}$$

- control constraints such as actuator limitations in amplitude or velocity:

$$U_{\min} \leq U(k) \leq U_{\max}. \tag{20.11}$$

These constraints are added to the problem (20.4) which becomes a nonlinear constrained optimization problem:

$$\min_{\tilde{U} \in \mathbb{K}} J(U) \tag{20.12}$$

where $\mathbb{K}$ is the constraint domain defined by

$$\begin{cases} C(U) \leq 0 \\ Ceq(U) = 0. \end{cases} \tag{20.13}$$

The constraints (20.9), (20.10) and (20.11) can be formulated by nonlinear functions $C(U)$ and $Ceq(U)$ [8]. Numerous constrained optimization routines are available in software libraries to solve this kind of problem: projected gradient methods, penalty methods, *etc*. In our case, a sequential quadratic program (SQP) is used and more precisely, the function *fmincon* from Matlab optimization toolbox.

The setting parameters of the predictive approach are the prediction horizon (N_p), the control horizon (N_c) and the weighted matrix ($Q(j)$):

- the prediction horizon is chosen in order to satisfy a compromise between scheme stability (long horizon) and numerical feasibility in term of computational time requirement (short horizon);
- the control input is usually kept constant over the prediction horizon, which corresponds to a control horizon equal to 1. A $N_c > 1$ can be useful for stabilization task of nonholonomic mobile robot for instance [1]; and
- the matrix $Q(j)$ is often the identity matrix but it can also be a time-varying matrix useful for stabilizing the system. If $Q(k+1) = I$ and $Q(k+l) = 0 \ \forall l \in [2; N_p]$, the cost function J is then similar to the standard criterion of IBVS. It is also equivalent to have a prediction horizon equal to 1.

20.3 Model of Image Prediction

Here we focus on the model used to predict the image evolution. We consider a 6 DOF free-flying perspective camera observing fixed point features. A 3D point with coordinates $P = (X, Y, Z)$ in the camera frame is projected in the image plane as a 2D point with coordinates $s = (u, v)$. The sampling period is T_e and the control input U is the camera velocity noted $\tau = (T_x, T_y, T_z, W_x, W_y, W_z)$.

The role of the model is to predict, over the horizon N_p, the evolution of the visual features in regard to the camera velocity. The principle of the image prediction is depicted in Figure 20.2. To perform this image prediction, two kinds of model can be considered: a nonlinear global model and a local model based on the interaction matrix. The identification of the model, described above by (20.7), is discussed with respect to both cases in the next section.

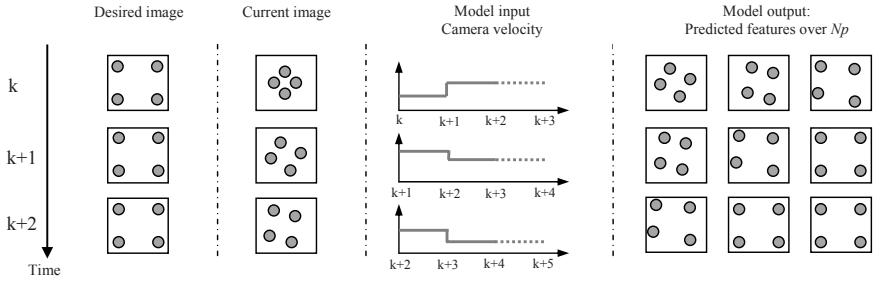


Fig. 20.2 Principle of image prediction $(N_p = 3, N_c = 2)$

20.3.1 Nonlinear Global Model

The control input of the free-flying process is the camera velocity τ applied to the camera. Here, the state of the system can be the camera pose in the target frame: $x = (P_x, P_y, P_z, \Theta_x, \Theta_y, \Theta_z)$. The dynamic equation can be approximated by[1]

$$x(k+1) = x(k) + Te\ \tau(k) = f(x(k), \tau(k)). \tag{20.14}$$

The output is the visual features expressed in the image plane noted s_m. In the case of a perspective camera, the output equation for one point-like feature in normalized coordinates can be written as

$$s_m(k) = \begin{pmatrix} u(k) \\ v(k) \end{pmatrix} = \begin{pmatrix} X(k)/Z(k) \\ Y(k)/Z(k) \end{pmatrix} = g(X(k), Y(k), Z(k)), \tag{20.15}$$

where $(X, Y, Z, 1)_{R_c}$ are the point coordinates in the camera frame. The rigid transformation between the camera frame and the target frame, noted $l(x)$, can easily be deduced from the knowledge of the camera pose $x(k)$. If the point coordinates are known in the target frame, $(X, Y, Z, 1)_{R_t}$, then the point coordinates in the camera frame, $(X, Y, Z, 1)_{R_c}$ are given by

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{R_c} = \begin{pmatrix} R(x) & T(x) \\ 0_{1\times 3} & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}_{R_t} = l(x(k)). \tag{20.16}$$

Finally, we obtain

$$s_m(k) = g \circ l(x(k)) = h(x(k)). \tag{20.17}$$

Equations 20.7 are now completely identified with (20.14) and (20.17). This dynamic model combines 2D and 3D data and so it is appropriate to deal with 2D and/or 3D constraints. The constraints are respectively expressed on the states and/or the outputs of the prediction model and are easily added to the optimization

[1] The exponential map could be also used to better describe the camera motion.

problem. The nonlinear global model has a large validity domain and thus it can be used for large displacements. Nevertheless, the prediction over the prediction horizon can be time consuming. Moreover, this model requires 3D data that are the pose of the target in the initial camera frame, as well as the target model. To reduce the 3D knowledge, a solution can be the linearization of the model based on the interaction matrix.

20.3.2 Local Model Based on the Interaction Matrix

For a point-like feature s expressed in normalized coordinates such that $u = X/Z$ and $v = Y/Z$, the interaction matrix related to s is given by [5]

$$L_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{u}{Z} & uv & -(1+u^2) & v \\ 0 & -\frac{1}{Z} & \frac{v}{Z} & 1+v^2 & -uv & -u \end{bmatrix}. \tag{20.18}$$

The value Z is the depth of the 3D point expressed in the camera frame. The relationship between the camera velocity τ and the time variation of the visual features $\dot{s}$ is given by

$$\dot{s}(t) = L_s(t)\tau(t). \tag{20.19}$$

In [11], this dynamic equation is solved to reconstruct the image data in case of occlusion. Here, with a simple first order approximation, we obtain

$$s(k+1) = s(k) + T_e\, L_s(k)\tau(k). \tag{20.20}$$

To avoid the estimation of the depth parameter at each iteration, its value Z^* given or measured at the reference position can be used. Consequently, the interaction matrix (20.18) becomes $\widehat{L_s}$ and depends only on the current measure of the visual features. By considering here the visual features s as the state x, we obtain the set of equations describing the process dynamics and outputs (20.7):

$$\begin{cases} x(k+1) = x(k) + T_e\, \widehat{L_s}(k)\tau(k) \;=\; f(x(k), \tau(k)) \\ s_m(k) = x(k) \;=\; h(x(k)). \end{cases} \tag{20.21}$$

This approximated local model does not require 3D data but only the approximate value of Z^*. 2D constraints can be taken into account since the model states and outputs are the visual features. On the other hand, no information is available on the camera pose and so 3D constraints can not be directly handled. For doing that, as for the nonlinear model, it would be necessary to reconstruct the initial camera pose by using the knowledge of the 3D model target. That is of course easily possible but has not been considered in this chapter. Finally, for large displacements, a problem can be mentioned as we will see on simulations: the linear and depth approximations may be too coarse and can lead to control law failures.

20.4 Simulation Results

For all presented simulations, the sampling time T_e is equal to 40 ms. This choice allows considering real-time application with an usual camera (25 fps). The control task consists in positioning a perspective free-flying camera with respect to a target composed of four points. These four points form a square of 20 cm in length in Cartesian space. The reference image is obtained when the target pose expressed in the camera frame (R_C) is equal to $P_{T/C} = (0,0,0.5,0,0,0)$ (see Figure 20.3), where the first three components are the translation expressed in meters and the last three components are the roll, pitch and yaw angles expressed in radians. The coordinates of the four points in the reference image are: $s^* = (u_{d1}, v_{d1}, u_{d2}, v_{d2}, u_{d3}, v_{d3}, u_{d4}, v_{d4}) = (-0.2, 0.2, 0.2, 0.2, 0.2, -0.2, -0.2, -0.2)$ (see Figure 20.4).

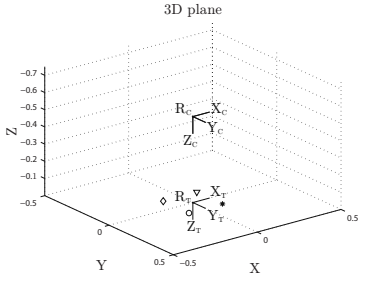


Fig. 20.3 3D desired posture

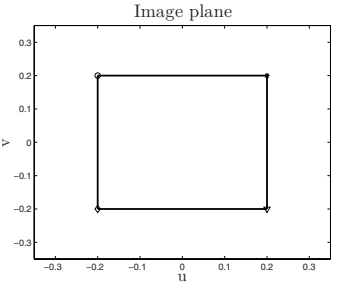


Fig. 20.4 2D reference image

Different simulations illustrate the performance of the VPC strategy. Besides the choice of the model used to predict the point positions, VPC requires to set three parameters, the prediction horizon N_p, the control horizon N_c and the weighted matrix $Q(j)$:

- the choice of the prediction horizon is crucial. The system behavior and the convergence speed depend on the prediction horizon. The value of N_p is discussed below;
- the control horizon is kept constant and equal to 1 ($N_c = 1$). Only one control is calculated over the prediction horizon; and
- the weighted matrix $Q(j)$ is either the identity matrix $Q(j) = I_{8\times 8}\ \forall j$, constant over the prediction horizon, or a time-varying matrix $Q(j) = 2Q(j-1)$ with $Q(1) = I_{8\times 8}$. In this last case, this matrix weights the error at each sampling instant more and more over the prediction horizon and so, stresses the error at the end of the horizon N_p which corresponds to the final objective. In stabilization task, this time-varying matrix can be compared to the terminal constraint used in the classical predictive control strategy. However it is less restrictive for the optimization algorithm. In the sequel, the time-varying matrix is noted Q_{TV}.

The VPC simulation results are compared with the classical IBVS approaches based on the exponential control law (20.1) where $\widehat{L_s}$ can be chosen as:

- $\widehat{L}_s = \widehat{L}_{(s(t),Z(t))}$, noted L_c: the interaction matrix is updated at each iteration;
- $\widehat{L}_s = \widehat{L}_{(s(t),Z^*)}$, noted L_p: the depth computed or measured at the reference position noted Z^* is used. The interaction matrix varies only through the current measure of the visual features;
- $\widehat{L}_s = \widehat{L}_{(s^*,Z^*)}$, noted L_d: the interaction matrix is constant and corresponds to its value at the reference position; and
- $\widehat{L}_s = \frac{1}{2}\left(\widehat{L}_{(s^*,Z^*)} + \widehat{L}_{(s(t),Z(t))}\right)$, noted L_m: the interaction matrix is the mean of the constant and current interaction matrices.

In order to compare the VPC approach with the classical IBVS, no constraint on the control input is considered in the first part. Then, mechanical and visibility constraints will be taken in consideration with the VPC approach. In all cases, the control inputs are normalized if needed. The bounds are 0.25 m/s for the translation speed and 0.25 rad/s for the rotation speed.

20.4.1 Case 1: Pure Rotation around the Optical Axis

In case 1, the required camera motion is a pure rotation of $\frac{\pi}{2}$ radians around the optical axis. Due to the lack of space and since it is a classical case, all simulation results are not presented here but all are discussed.

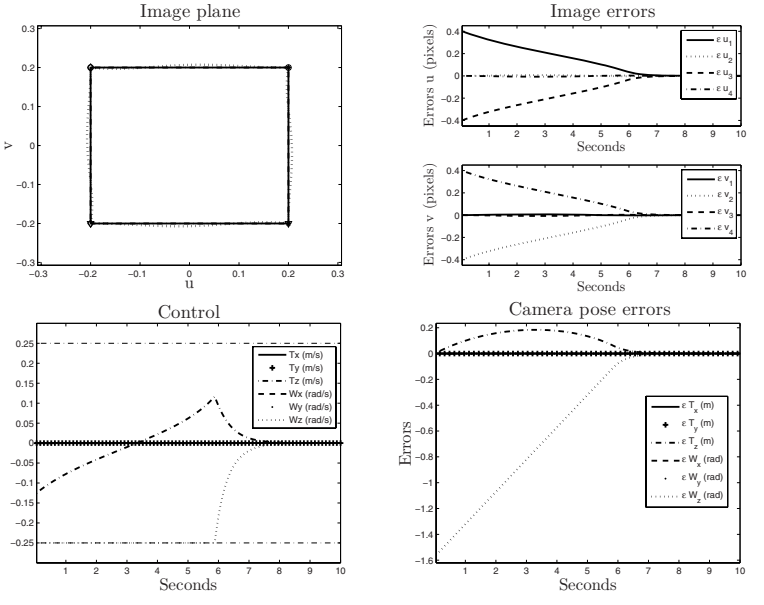


Fig. 20.5 Case 1: Classical IBVS with L_p

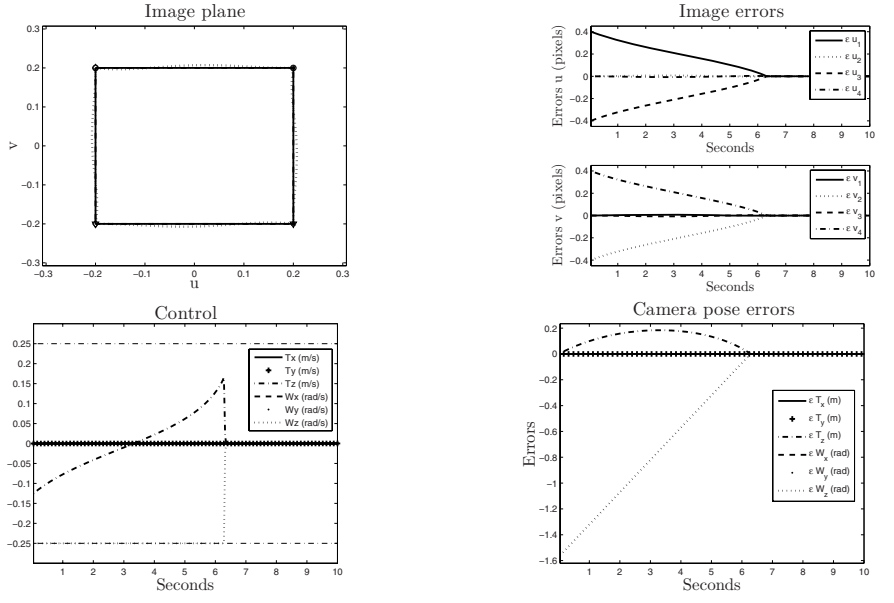


Fig. 20.6 Case 1: VPC_{LM} with $N_p = 1$, $Q(j) = I$

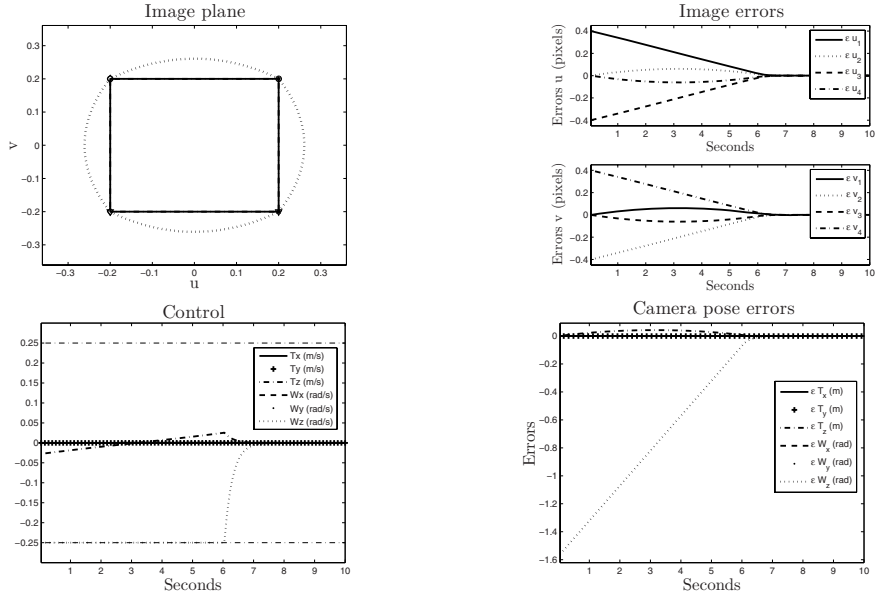


Fig. 20.7 Case 1: VPC_{LM} with $N_p = 10$, $Q(j) = I$

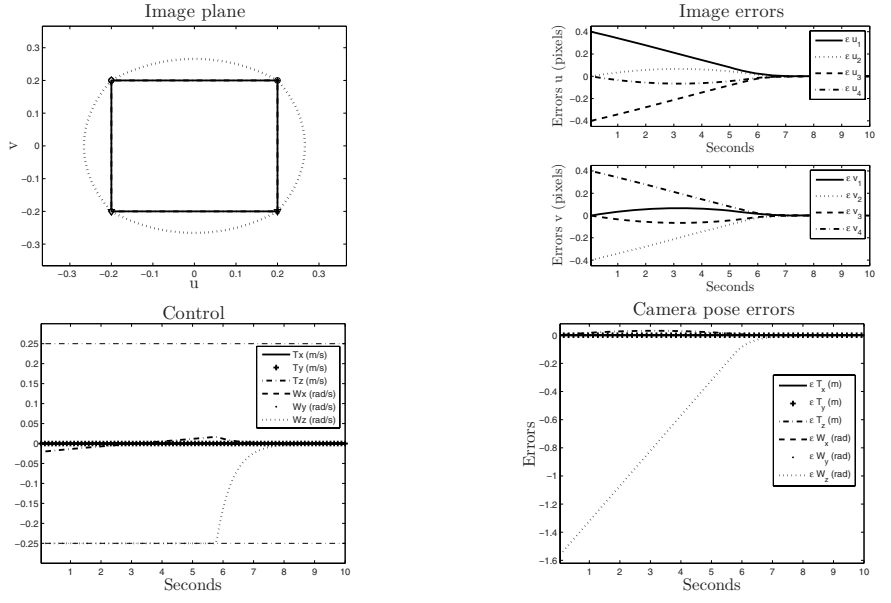


Fig. 20.8 Case 1: VPC_{LM} with $N_p = 20$, $Q(j) = I$

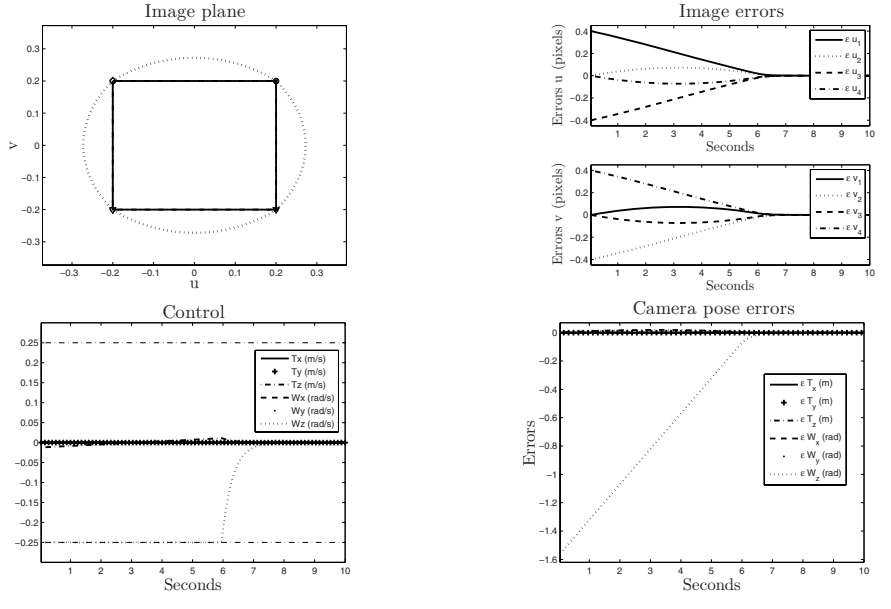


Fig. 20.9 Case 1: VPC_{LM} with $N_p = 10$, $Q(j) = Q_{TV}$

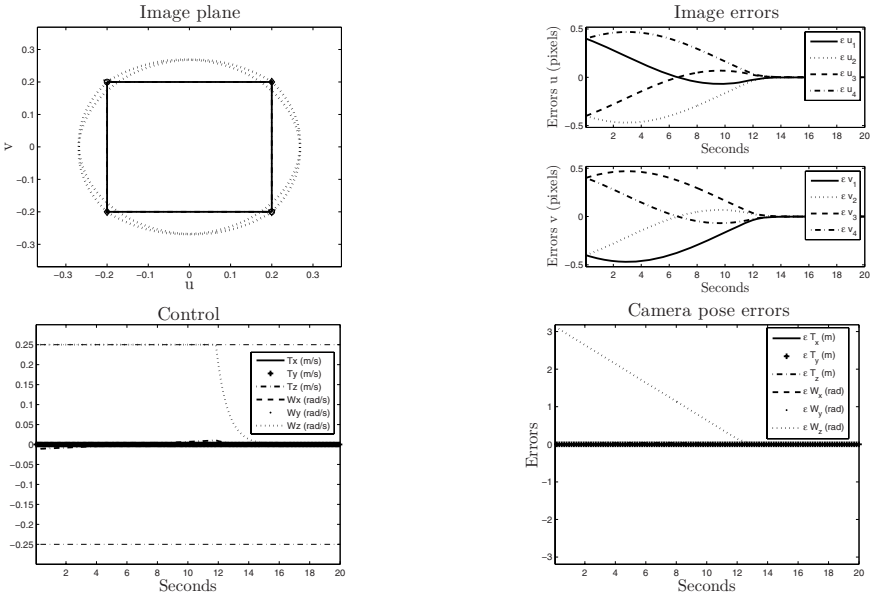


Fig. 20.10 π rotation around optical axis: VPC_{LM} with $N_p = 20$, $Q(j) = Q_{TV}$

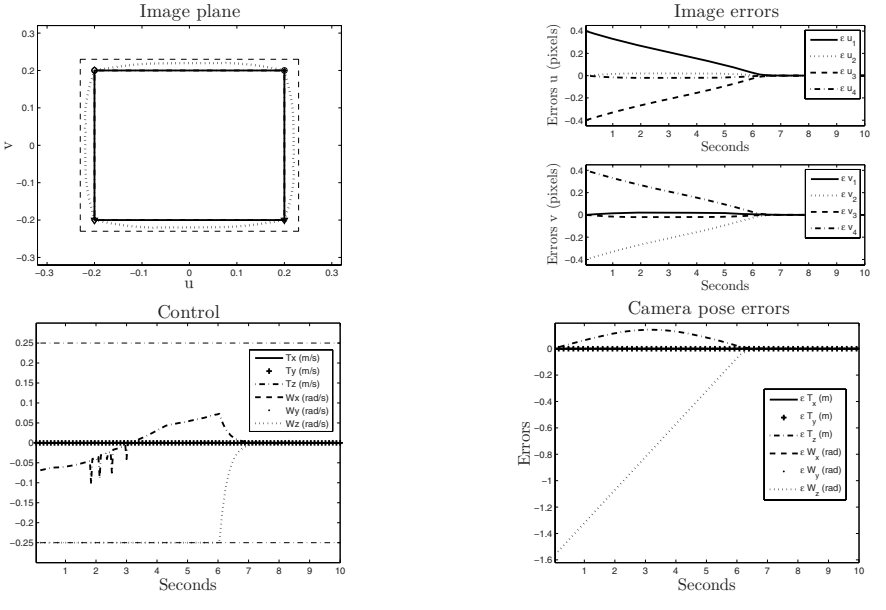


Fig. 20.11 Case 1: VPC_{LM} with $N_p = 10$, $Q(j) = I$ and visibility constraint

20.4.1.1 Classical IBVS

For the classical IBVS, the following results are obtained:

- with L_c, the trajectories in the image plane are pure straight lines as expected [5]. The camera motion is then a combination of a backward translation and a rotation with respect to the camera optical axis (retreat problem). Due to this undesired retreat, the camera might reach the limit of the workspace;
- with L_p, the results are approximatively the results obtained in the first case (retreat problem), see Figure 20.5. The visual feature trajectories tend to straight lines;
- with L_d, the camera moves toward the target and simultaneously rotates around the optical axis (advance problem) [5]. Due to this undesired forward motion, some features can go out the camera field of view during the camera motion; and
- with L_m, the camera motion is a pure rotation [18]. No additional motion is induced along the optical axis and the visual feature trajectories are circular.

20.4.1.2 VPC with a Local Model ($\mathbf{VPC_{LM}}$)

The following simulations are obtained with the VPC strategy using a local model based on the interaction matrix L_p. The comparison with the classical IBVS is done for different N_p values ($N_p = 1, 10, 20$) and different weighted matrices ($Q(j) = I$ or Q_{TV}). For $N_p = 1$, the results are similar to the classical IBVS with L_p since the model used to predict the next image is exactly the same (see Figure 20.6). The only difference is the behavior of the control law, decreasing exponentially with IBVS. For $N_p = 10$ (see Figure 20.7) or $N_p = 20$ (see Figure 20.8), the trajectories in the image plane become circular. Indeed, the only constant control over N_p which minimizes the cost function is a pure rotation. Thus the translation motion along the optical axis decreases with the increase of N_p value.

The time-varying matrix Q_{TV} accentuates the decoupling control by giving importance at the end of N_p which corresponds to the final objective (see Figure 20.9). It seems to be equivalent to the behavior obtained with L_m which takes into account the desired position. For a π rotation around the optical axis, the classical IBVS with L_c, L_p or L_d fails as well as $\mathrm{VPC_{LM}}$ with $Q(j) = I$ and whatever N_p. On the other hand, $\mathrm{VPC_{LM}}$ achieves the satisfying motion with Q_{TV} and $N_p \geq 20$ (see Figure 20.10).

To illustrate the capability of visibility constraint handling, the visual features are constrained to stay in a window defined by the inequalities

$$\begin{bmatrix} \mathrm{u_{min=}} - 0.22 \\ \mathrm{v_{min=}} - 0.22 \end{bmatrix} \leq \mathrm{s_m}(j) \leq \begin{bmatrix} \mathrm{u_{max}} = 0.22 \\ \mathrm{v_{max}} = 0.22 \end{bmatrix}. \tag{20.22}$$

In that case, $\mathrm{VPC_{LM}}$ satisfies both visibility constraint and control task (see Figure 20.11). A translation along the optical axis is then induced to ensure that the visual features do not get out the camera field of view.

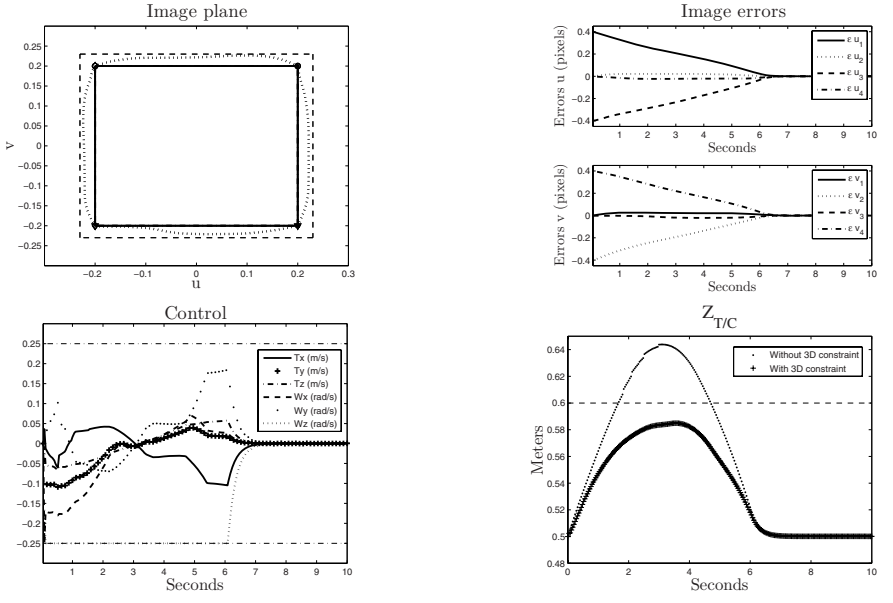


Fig. 20.12 Case 1: VPC_{GM} with $N_p = 10$, $Q(j) = I$, visibility constraint and $Z_{T/C} < 0.6$ m

20.4.1.3 VPC with a Nonlinear Global Model (VPC_{GM})

The previous results obtained with VPC_{LM} are improved with VPC_{GM} since no linearization is done. For instance, with VPC_{GM} and $N_p = 1$, the image plane trajectories are circular as the ones obtained with L_m. Here, we focus on 3D constraint handling. Added to the visibility constraint, we limit the camera workspace along the Z_c axis. As can be seen on Figure 20.12, VPC_{GM} converges under both constraints by using the other camera DOF. If no admissible trajectory ensuring visibility constraints and 3D constraints (such as $-0.05 < X_c < 0.05,\ -0.05 < Y_c < 0.05,\ Z_c < 0.6m$) exists, VPC_{GM} stops at the position minimizing the constrained cost function (see Figure 20.13).

20.4.2 Case 2: Large Displacement

In case 2, the initial target pose expressed in the camera frame is given by $P_{T/C} = (0.04, 0.03, 0.35, 1, -0.98, -0.43)$. The classical IBVS with L_p or L_d does not converge for such a large displacement. Indeed, during the motion, the camera reaches the object plane where $Z = 0$. The same result is obtained by the VPC_{LM} whatever N_p due to the too coarse approximations. However, the convergence is obtained with the weighted time-varying matrix Q_{TV}. VPC_{GM} always converges even if $Q(j) = I$ (see Figure 20.15). The trajectories in the image plane are very similar to the ones obtained with the classical IBVS with L_m (see Figure 20.14). These good results

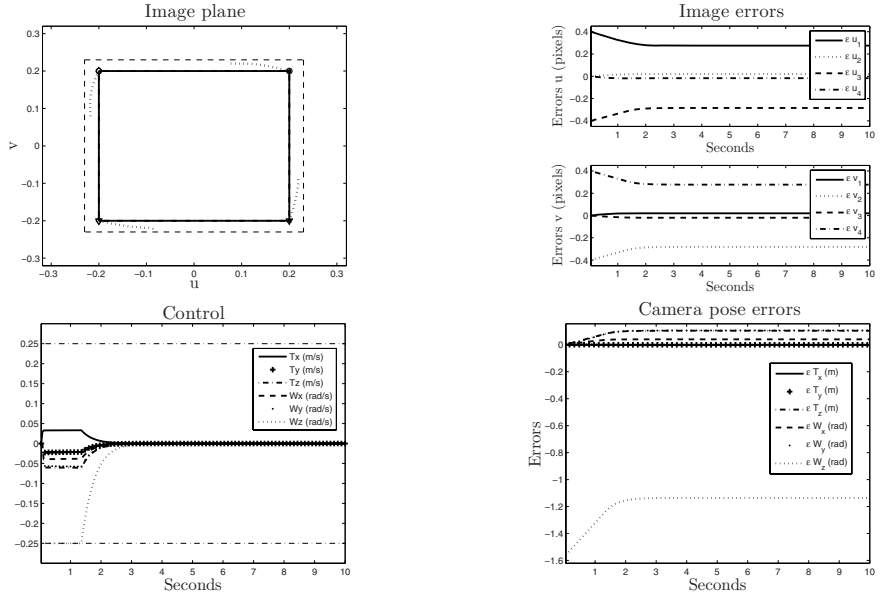


Fig. 20.13 Case 1: VPC_{GM} with $N_p = 10$, $Q(j) = I$, visibility and strong 3D constraints such that there is no solution to the optimization problem

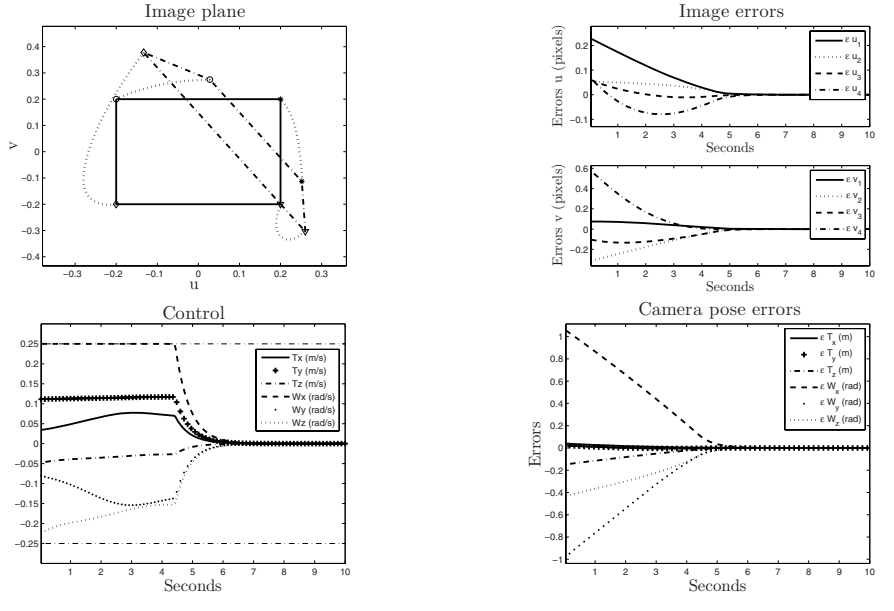


Fig. 20.14 Case 2: Classical IBVS with L_m

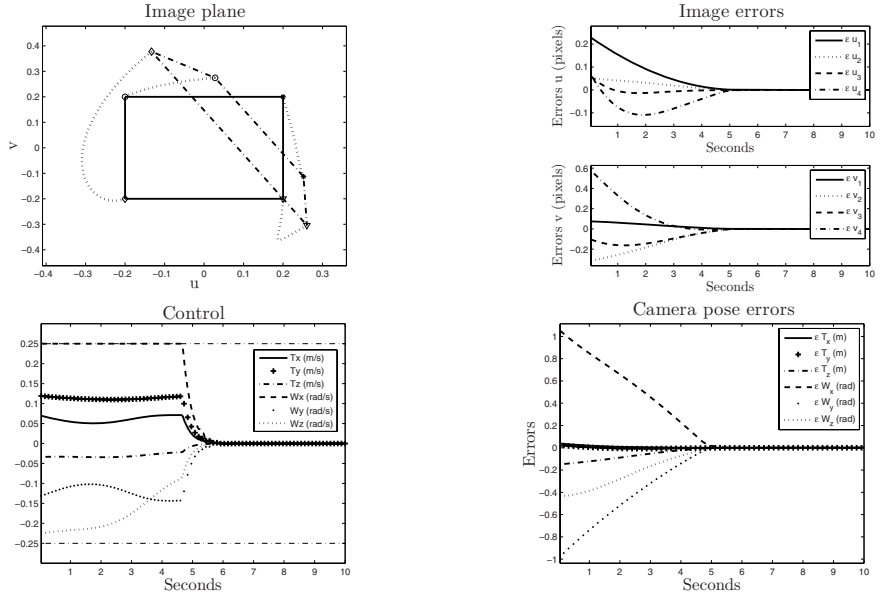


Fig. 20.15 Case 2: VPC_{GM} with $N_p = 10$, $Q(j) = I$

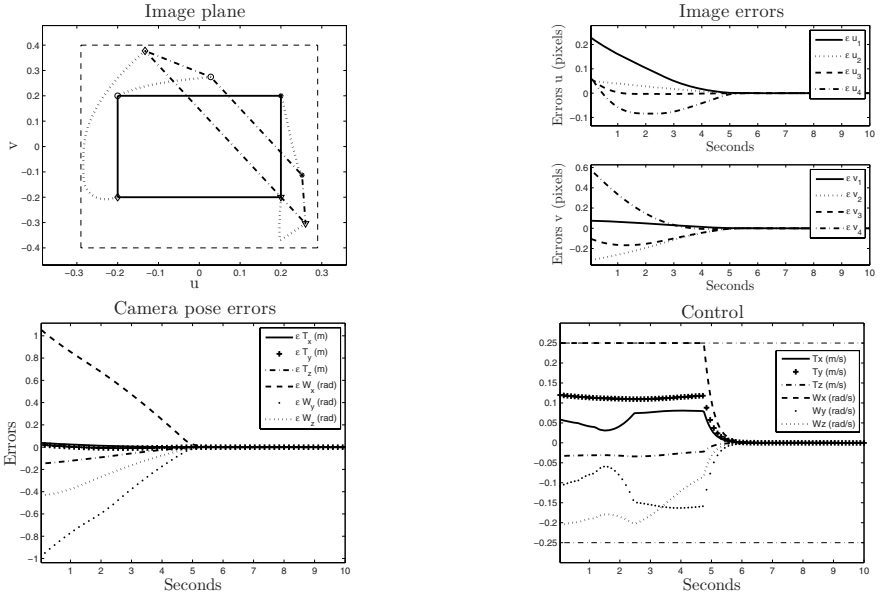


Fig. 20.16 Case 2: VPC_{GM} with $N_p = 10$, $Q(j) = I$ and visibility constraint

are still kept even if visibility constraints $(-0.29 < u < 0.29,\ -0.4 < v < 0.4)$ are considered (see Figure 20.16).

20.5 Conclusions

In this chapter, we have shown that an alternative approach of IBVS can be the VPC strategy. The visual servoing task is then formulated into a nonlinear optimization problem over a prediction horizon. The advantage of this formulation is the capability of easily dealing with visibility constraints and 3D constraints. The optimization procedure can be compared to an on-line implicit and optimal constrained path-planning of features in the image plane. The choice of the image prediction model has been discussed. The approximated local model can be less efficient than the global model for difficult configurations but no 3D data are required. On the other hand, if 3D data are available, VPC_{GM} gives satisfying results for any initial configuration and motion to achieve. The VPC setting parameters, *i.e.*, the prediction horizon and the weighted matrix, play a crucial role in terms of camera and visual feature trajectories. Simulation results highlight the efficiency of VPC. Finally, this strategy is very flexible and can be used whatever the robotic system (mobile robot or robot arm) and the camera (perspective or catadioptric).

References

[1] Allibert, G., Courtial, E., Touré, Y.: Visual predictive control. In: IFAC Workshop on Nonlinear Predictive Control for Fast Systems, Grenoble, France (2006)
[2] Allibert, G., Courtial, E., Touré, Y.: Real-time visual predictive controller for image-based trajectory tracking of mobile robot. In: 17th IFAC World Congress, Seoul, Korea (2008)
[3] Allibert, G., Courtial, E., Touré, Y.: Visual predictive control for manipulators with catadioptric camera. In: IEEE Int. Conf. on Robotics and Automation, Pasadena, USA (2008)
[4] Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman, D., Hager, G., Morse, A. (eds.) The Confluence of Vision and Control. LNCIS, vol. 237, pp. 66–78. Springer, Heidelberg (1998)
[5] Chaumette, F., Hutchinson, S.: Visual servo control, part i: Basic approaches. IEEE Robotics and Automation Magazine 13(4), 82–90 (2006)
[6] Chaumette, F., Hutchinson, S.: Visual servo control, part ii: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)
[7] Chesi, G.: Visual servoing path planning via homogeneous forms and lmi optimizations. IEEE Trans. Robotics and Automation 25(2), 281–291 (2009)
[8] Chong, E., Zak, S.H.: An Introduction to Optimization, 2nd edn. John Wiley & Sons Inc., Chichester (2001)
[9] Danès, P., Bellot, D.: Towards an lmi approach to multicriteria visual servoing in robotics. European Journal of Control 12(1), 86–110 (2006)
[10] Durola, S., Danès, P., Coutinho, D., Courdesses, M.: Rational systems and matrix inequalities to the multicriteria analysis of visual servos. In: IEEE Int. Conf. on Robotics and Automation, Kobe, Japan (2009)

[11] Folio, D., Cadenat, V.: Dealing with visual features loss during a vision-based task for a mobile robot. International Journal of Optomechatronics 2(3), 185–204 (2008)
[12] Gangloff, J., De Mathelin, M.: Visual servoing of a 6 dof manipulator for unknown 3-d profile following. IEEE Trans. Robotics and Automation 18(4), 511–520 (2002)
[13] Ginhoux, R., Gangloff, J., De Mathelin, M., Soler, M., Sanchez, L.: Active filtering of physiological motion in robotized surgery using predictive control. IEEE Trans. Robotics and Automation 21(1), 67–79 (2005)
[14] Hashimoto, K., Kimura, H.: LQ optimal and nonlinear approaches to visual servoing. In: Hashimoto, K. (ed.) Visual Servoing, Robotics and Automated Systems, pp. 165–198. World Scientific, Singapore (1993)
[15] Hutchinson, S., Hager, G.D., Corke, P.: A tutorial on visual servo control. IEEE Trans. Robotics and Automation 12(5), 651–671 (1996)
[16] Kazemi, M., Gupta, K., Mehrandezh, M.: Global path planning for robust visual servoing in complex environments. In: IEEE Int. Conf. on Robotics and Automation, Kobe, Japan (2009)
[17] Mahony, R., Corke, P., Chaumette, F.: Choice of image features for depth-axis control in image-based visual servo control. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Lausanne, Switzerland (2002)
[18] Malis E.: Improving vision-based control using efficient second-order minimization techniques. In: IEEE Int. Conf. on Robotics and Automation, New Orleans, LA, USA (2004)
[19] Mezouar, Y., Chaumette, F.: Optimal camera trajectory with image-based control. Int. Journal of Robotics Research 22(10), 781–804 (2003)
[20] Morari, M., Zafiriou, E.: Robust Control. Dunod (1983)
[21] Nasisi, O., Carelli, R.: Adaptive servo visual robot control. Robotics and Autonomous Systems 43(1), 51–78 (2003)
[22] Papanikolopoulos, N., Khosla, P., Kanade, T.: Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control. IEEE Trans. Robotics and Automation 9(1), 14–35 (1993)
[23] Sauvée, M., Poignet, P., Dombre, E., Courtial, E.: Image based visual servoing through nonlinear model predictive control. In: 45th IEEE CDC, San Diego, CA, pp. 1776–1781 (2006)
[24] Schramm, F., Morel, G.: Ensuring visibility in calibration-free path planning for image-based visual servoing. IEEE Trans. Robotics and Automation 22(4), 848–854 (2006)

Editors' Biographies

Graziano Chesi

Graziano Chesi received his Laurea in Information Engineering from the University of Florence in 1997, and his Ph.D. in Systems Engineering from the University of Bologna in 2001. From 2000 to 2006 he was with the Department of Information Engineering of the University of Siena, and in 2006 he joined the Department of Electrical and Electronic Engineering of the University of Hong Kong. He was a visiting scientist at the Department of Engineering of the University of Cambridge from 1999 to 2000, and at the Department of Information Physics and Computing of the University of Tokyo from 2001 to 2004.

Dr. Chesi was the recipient of the Best Student Award of the Faculty of Engineering of the University of Florence in 1997. He was Associate Editor of the *IEEE Transactions on Automatic Control* from 2005 to 2009, and Guest Editor of the Special Issue on *Positive Polynomials in Control* of the *IEEE Transactions on Automatic Control* published in 2009. Since 2007 he serves as Associate Editor for *Automatica*. He is the first author of more than 90 technical publications, including the book *Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems* (Springer). He is the Founder and Chair of the Technical Committee on Systems with Uncertainty of the *IEEE Control Systems Society*. His research interests include automatic control, computer vision, nonlinear systems, optimization, robotics, systems biology, and uncertain systems.

Dr. Graziano Chesi
Department of Electrical and Electronic Engineering
University of Hong Kong
Pokfulam Road, Hong Kong
Phone: +852-22194362
Fax: +852-25598738
Email: chesi@eee.hku.hk
Homepage: http://www.eee.hku.hk/~chesi

Koichi Hashimoto

Koichi Hashimoto is a Professor at the Graduate School of Information Sciences, Tohoku University. He received his BS, MS and DE degrees from Osaka University in 1985, 1987 and 1990, respectively. His major research interests include visual servoing, parallel processing and biological systems.

Prof. Hashimoto is a member of the IEEE, RSJ, SICE, ISCIE, IPSJ and JSME. He is Editor of the books *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback* (World Scientific) and *Control and Modeling of Complex Systems* (Birkhauser).

Prof. Koichi Hashimoto
Department of System Information Sciences
Tohoku University
6-6-01 Aramaki-Aza Aoba, Aoba-ku, Sendai 980-8579, Japan
Phone: +81 22 7957018
Email: koichi@ic.is.tohoku.ac.jp
Homepage: http://www.ic.is.tohoku.ac.jp/en/koichi

Index

3D reconstruction, 4, 43, 164, 195, 210, 296, 366, 382

amplitude modulation, 100
Armijo rule, 257

biography, editors', 395
Bryant angles, 171

camera
  calibration, 4, 61, 120, 170, 190, 212, 231, 297, 338
  catadioptric, 4, 210, 233, 377
  fisheye, 233, 305
  omnidirectional, 233, 336
  unified model, 233, 296
Cayley representation, 157, 197
collineation, 118, 194, 298
collision, 192, 316
column-parallel vision system, 42
coronary artery bypass grafting, 91

decoupled control, 192, 232, 302
differential algebraic representation, 177
differential drive robot, 198
domain of attraction, 29, 174, 276

efficient second order minimization, 79, 100, 146
eigenvalue problem, 159
electrocardiogram, 97
epipolar geometry, 7, 118, 193, 298, 352
essential matrix, 164, 298
evolution robotics software platform, 347

Fourier linear combiner, 98
Frobenius norm, 136
fundamental matrix, 8, 352

Gauss–Newton method, 79, 102
generalized predictive control, 105
global positioning system, 216
gradient projection method, 254
grasp, 40
Gronwall–Bellman lemma, 218

high-speed control, 39, 94
homography, 73, 118, 133, 193, 212, 296, 338
human-machine interaction, 55

illumination, 74
image intensity, 72
image moment, 232
inertial measurement unit, 216
internal model control, 378

Jacobian matrix, 24, 62, 76, 172, 237, 255, 303, 365, 376

Kalman filter, 210, 330, 342
kinematics, 25, 57, 134, 171, 192, 277, 323, 336

LaSalle theorem, 138, 344
Levenberg–Marquardt method, 79, 102, 353
Lie algebra, 136
linear matrix inequality, 152, 170, 197, 276, 376

linear parameter varying, 100
localization, 315
luminance, 72
Lyapunov function, 24, 134, 174, 211, 276
  biquadratic, 177
  piecewise-biquadratic, 178

magnetic resonance imaging, 91
Mahalanobis distance, 330
manipulation, 40, 55, 92
maximally stable extremal regions, 100
measurement error, 152
minimally invasive surgery, 92
mirror, 4, 233
mobile robot, 95, 276, 316, 337, 377
model predictive control, 376
Moore–Penrose inverse, 323, 366
motion from structure, 209

navigation, 315
Newton–Raphson method, 126

observer, 134, 210
occlusion, 24, 85, 192, 266, 336, 361, 379
odometry, 289, 337
optical flow, 74
organization for economic cooperation and development, 91

path-planning, 118, 152, 190, 376
perceptual control manifold, 196
PID controller, 45, 368
planar catadioptric stereo, 4
planar object, 25, 73, 133, 211, 232
polytopic representation, 170, 276
pose estimation, 5, 164, 170, 209, 233
positioning error, 152
potential field, 193, 258, 319

rational system, 170
recursive least-squares, 98
redundancy, 86, 254
robot arm, 4, 46, 56, 105, 118, 232, 254, 316, 377
robotic finger, 49
Rodrigues formula, 197, 243

S-procedure, 176
saturation, 276
scale invariant feature transformation, 209
semidefinite programming, 159, 176
simultaneous localization and mapping, 337
square matrix representation, 152
steady-state error, 37, 152
structure from motion, 209
surgery, 56, 92
switched control, 118, 192, 337
Sylvester equation, 15

teleoperation, 55, 94
texture, 100
totally endoscopic coronary artery bypass, 92

ultrasound, 290
unicycle-like robot, 316, 337
unmanned aerial vehicle, 361

variational problem, 118
virtual fixture, 57
visual predictive control, 376
visual servoing
  featureless, 24, 100
  kernel-based, 23, 73
  photometric, 72
  virtual, 233

Zeno phenomenon, 344
zero-order hold, 106