Marilyn Wolf

Smart Camera Design

Algorithms, Architectures, and Art

Springer

Smart Camera Design

Marilyn Wolf

Smart Camera Design

Algorithms, Architectures, and Art

Springer

Marilyn Wolf
School of ECE
Georgia Institute of Technology
Atlanta, GA, USA

ISBN 978-3-319-69522-8 ISBN 978-3-319-69523-5 (eBook)
https://doi.org/10.1007/978-3-319-69523-5

Library of Congress Control Number: 2017957575

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

For Dad,
Who set me on this path.

Preface

This book draws from three areas of computing: image processing, computer vision, and computer graphics. Image processing and computer vision in particular have long been separate fields with overlapping interests. This is partly a sociological phenomenon—image processing comes from electrical engineering, while computer vision comes from computer science. These two fields blend smoothly in digital camera design. The modern imaging chain starts at traditional filtering and ends with feature analysis.

Parts of this book draw upon my research work with my students at Princeton and Georgia Tech. Cheng-Yao Chen, Santanu Dutta, Jason Fritts, Se Hun Kim, Changhong Lin, Chung-Ching Lin, Tiehan Lv, Jason Schlessman, Senem Velipasalar, Jiang Xu, Heather Yu, and Shengqi Yang have all worked on aspects of multimedia computing and embedded computer vision. I am grateful to them for the opportunity to work with them and learn from them. Burak Ozer was not my official student, but he has been my friend and collaborator on smart cameras for the past 15 years.

The inspiration for this book comes from my father, an inventor who created two different panoramic cameras. The camera on the dedication page, which was known commercially as the CycloPan 360, uses a cylindrical mapping. Here is another picture of Dad with his donut camera.

He also created a motion picture version of this camera for use in a flight simulator. That camera used 5″ aerial film. The projector used the same optical path as the camera but rotated continuously to sweep the image; it was an imposing machine. I ran Dad's color darkroom during those years—I worked cheap.

Dad, this book is entirely your fault. I have spent night after night typing away because you taught me how to think. This book is my tribute to you.

Atlanta, GA, USA Marilyn Wolf
August 2016

Contents

Chapter 1
Digital Photography

1.1 Introduction

Photography has evolved considerably in the two centuries since its invention. Advances have allowed us to take more sophisticated, accurate photos with less technical knowledge. The introduction of semiconductor image sensors and embedded processors has formed the foundation for the latest set of advances. This book studies the range of technologies that have enabled us to build *smart cameras*. The move to smart cameras enables three trends, each of which we will analyze through the book and introduce in this chapter: previsualization and autoprevisualization, automated enhancement of photographs, and cameras that produce analytical summaries rather than photographs.

1.2 Previsualization and Autoprevisualization

Digital cameras have changed the face of photography. Both casual and professional photographers can now take better pictures with less effort than ever before. The goal of this book is to outline the technologies that make this possible.

Better pictures require good algorithms. But those algorithms ultimately must make decisions about how to process the picture to get the "best" result. And "best" is clearly a subjective criterion. I believe that at the heart of the digital camera revolution is the move from *previsualization* by the photographer to *autoprevisualization* by the camera. Previsualization is a term introduced by Ansel Adams [Ada02A, Ada02B, Ada02C]—he taught photographers to see in their mind's eye how they wanted their photo to look and then determine the proper combination of techniques to achieve that result. Previsualization is a human, artistic endeavor. Cameras cannot make the sort of profound artistic judgments that Ansel Adams did, but they can make choices based on scene characteristics and

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5_1

knowledge about the composition of typical photographs. The result is autoprevisualization, an automation of the photographic art. Today's cameras are not gallery-ready artists, but they often make better decisions than do typical snapshot shooters.

We need some sort of previsualization because photographs require careful construction to give us a useful and interesting representation of a scene. The human visual system operates on profoundly different principles than does a camera. We perform a great deal of processing when we look at something without being the slightest bit aware—our eye constantly scans, constantly adjusts for focus and exposure, and continually identifies objects of interest. Capturing an image and looking at the resulting photo are two very different experiences.

Technical applications of digital cameras need previsualization even more. Autonomous automobiles give just one example. These cars rely on cameras to identify both roads and obstacles. These cameras must work reliably under a huge range of environmental conditions. The car's cameras must be able to adjust themselves continually to deliver the information required to safely drive the car—the driver cannot twiddle the knobs to keep the vision system working.

Photographic technology has steadily moved toward simpler processes since its earliest days. The Kodak, a simple box camera made possible by the advent of roll film, helped to establish the snapshot as a tradition; professional photographers were no longer needed to take photos. Film cameras started to add exposure mechanisms in the 1960s and autofocus in the 1970s. But digital image sensors allowed cameras to analyze images before, during, and after capture, making possible a much broader spectrum of optimizations and interventions into the photographic process.

1.3 Enhanced Images

Cameras are physical devices. A number of factors constrain the photograph we can capture of a given scene: lighting, camera position, optics, and sensor characteristics. Film photography gave us some tools with which to manipulate photographs to enhance the image. A photographer could, for example, dodge and burn parts of the print in order to alter the contrast within the image. Digital photography gives us a much broader range of options. Early tools for image manipulation and enhancement naturally emulated the techniques and results of film photography. Increasingly, digital techniques allow us to create images that simply were not possible with film. Focus stacking, for example, allows us to combine several photos in order to create a composite with much greater depth-of-field. High-dynamic range (HDR) algorithms allow us to combine photos with different exposures to create a composite that re-renders the lighting of the scene.

1.4 Beyond Images to Analysis

Computer vision algorithms allow us to move beyond producing a photograph at all. Cameras are widely used to identify people and objects or to analyze and classify their activity. Analysis has some advantages over imaging—when cameras are used for safety and security, many people are more comfortable knowing that images do not leave the camera. Algorithms can also combine information from multiple cameras to create an even more accurate and complete understanding of a scene. Multiple cameras reduce occlusion and can also provide several views of a subject at multiple resolutions and perspectives.

1.5 Still and Moving Images

One of the interesting side effects of the digital camera resolution is a blurring of the traditional boundary between still and motion picture cameras. In the film era, the two were very different beasts. In the digital era, the differences between the two become much smaller. Virtually all cameras today have some capability to capture both still and moving images—they may be better at one than the other, but they can do both. This book will move fluidly between still and video.

1.6 Taking a Picture

To understand just how much modern cameras do for us, let us consider the picture-taking process. The photograph of Fig. 1.1 is not complicated or a work of art, merely an enjoyable photo. Yet even taking this simple photo required some care and consideration.

First, the steps that take place before a still photo are taken:

- The camera is positioned to have a chosen view of the subject. The position includes not only the x, y, z position of the camera but also its orientation.
- The image is focused on a particular part of the subject.
- The required exposure is determined.

Once the photo is actually captured, the camera performs a number of steps, some of which may be optional depending on the sophistication of the camera or the choices made by the photographer:

- The scene's white balance is determined to compensate for the different colors produced by different types of light sources.
- The image may be sharpened to make it more pleasing.

Fig. 1.1 An uncomplicated photograph

- The image data is compressed, typically with lossy algorithms that throw away some aspects of the image in order to reduce the amount of data required to reproduce the image.
- The compressed image data is stored as a file in a storage medium.

The process for video is much the same except that most of these steps must be performed continually: focus, exposure, image enhancement, compression, and storage all require *streaming* operation.

Many of these operations require some sort of judgment—there is no single answer as to what makes the best picture in most situations. Given the high degree of automation of today's cameras, you may not have thought much about some of those decisions:

- What should you focus on? Is the subject of the photograph the flower near the camera or the mountain far away?
- What exposure should you use? A person is standing in front of a bright window. You and your camera have two choices: the background is properly exposed and clearly visible, leaving the person dark and unintelligible; or the background is blown out and the person is clearly distinguishable.

- How much should you compress this photo? Do you care more about file size or image detail? Is this a snapshot or a technical photograph?

Algorithms can help us with many of these tasks. (They cannot help us take our photos on a perfect Utah evening unless we rely on surveillance cameras that continually monitor everywhere and everything.) But algorithms can help us take better photos and videos:

- Early-stage operations such as autofocus, autoexposure, and auto white balance
- Image enhancements such as sharpening and keystone correction or, in the case of video, stabilization
- Composite photographs such as high-dynamic range (HDR) and mosaics

This book is intended to walk through the major operations in digital photography and to understand the trade-offs in the design of camera systems.

1.7 How to Read this Book

This book was written to address a range of readers who may have diverse backgrounds. I believe that the topics in the book are important for a full understanding of smart camera design, but not everyone may have the same depth of interest in all of these topics. I have tried to arrange the subsections within sections so that the major concepts of a section can be grasped without necessarily resorting to all the necessary details.

All technical people interested in digital cameras and photography should, in my opinion, have at least a basic appreciation of the arts of photography and cinematography. Over the years, I have found the fields of computer music vs. image processing and computer vision to be populated by very different types of people. Computer music people are invariably musicians who have a deep, intuitive sense of what they want to accomplish with their designs. Image processing and computer vision specialists, in contrast, rarely have even a basic understanding of the photographic arts. I think that an appreciation of how we use photos is at the heart of autoprevisualization and essential to a truly in-depth understanding of digital camera design. A corollary is that a fair amount of the technical material required to understand digital camera design is not unique to digital. Optics and the physics of light still apply in the digital domain.

The chapters are designed to explore different aspects of digital cameras:

- Chapter 2, *Light, Optics, and Imaging*, examines how images are formed and displayed. It looks at the nature of light, optics, and the human visual system. It also considers the more practical aspects of image capture, leading to a discussion of previsualization.
- Chapter 3, *Image Capture Systems and Algorithms*, studies the design of cameras as machines. We consider optics, image sensors, cameras as multiprocessors, and the basic operations required to automate the photographic process. We also

study the basic algorithms in the imaging path, such as sharpening and compression.

- Chapter 4, *Image and Video Enhancement*, studies algorithms for more advanced image and video operations, such as high-dynamic range and image mosaicing.
- Chapter 5, *Image and Video Analysis*, considers algorithms that analyze imagery and video with the goal of reducing the images to their characteristics: scene recognition, tracking, etc.
- Chapter 6, *Photography and Cinematography*, surveys the arts of photography and cinematography. We look at some of the major styles and approaches to these arts. We also consider how technological changes over the nearly 200-year history of photography have influenced the practice and range of the art.

Chapter 2
Light, Optics, and Imaging

2.1 Introduction

Digital cameras capture images; we need to understand how to form and control images before we can fully understand how digital cameras work. This chapter sets the stage by identifying several key photographic problems that we will solve in the succeeding chapters. We will start with a basic understanding of image formation based on pinhole cameras. We will then survey the human visual system in Sect. 2.3 and then study color more deeply in Sect. 2.4. Sections 2.5 and 2.6 concentrate on the basics of imaging, both lenses proper and a simpler model of the camera. Section 2.7 briefly discusses image display. The final section integrates this material into a practical view of image capture. We will discuss *previsualization* as a technique to help people figure out how to take the image they want. That discussion will set us up to understand *autoprevisualization* methods in the next two chapters. Nothing in this chapter is specific to digital cameras—a camera is a camera.

2.2 Image Formation

Image formation is the starting point for image capture. A few words on imaging and the physics of light help to inform our later discussions.

2.2.1 Light and Images

Images do not just happen. We need to use devices to control light that allows the formation of an image that we can see. Some very simple examples show what we mean by image formation.

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5_2

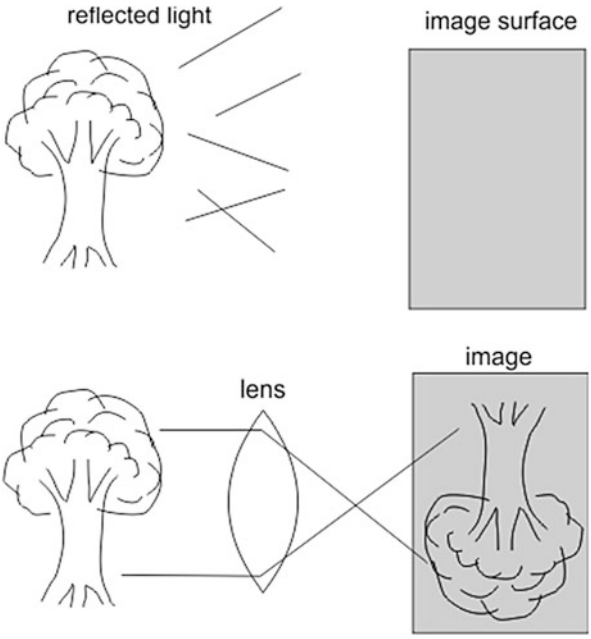


Fig. 2.1 Image formation

Figure 2.1 shows a subject and an *image surface* on which we want to throw an image of the subject. In natural scenarios, light comes to a subject from many different directions. That light is reflected from the subject in many directions. Similarly, each point on the imaging surface receives light from many different points in the scene. By using a device—a *lens*—to control how light reaches the imaging surface, we can ensure that the light at each point on the imaging surface comes from a single point on the subject (or at least from a small area on the subject). The relationship of the camera to the subject is known as the *point of view* (*POV*). The result is an image of the subject. This configuration is known as a camera obscura and has been used for centuries to project images. The camera obscura's images are, however, transitory phenomena. The invention of photography allowed us to capture and preserve these images.

An even simpler approach to image formation uses a *pinhole camera*—this toy is also a useful abstract model for the camera. As shown in Fig. 2.2, the pinhole restricts the paths to points in the image plane such that an image can form. The size of the pinhole has several effects on the quality of the image. On the one hand, a smaller pinhole gives us a clearer, more focused picture. A larger pinhole allows a cone of light from the subject to fall onto a point on the imaging surface, producing a circle of light. The circles from different points on the subject overlap on the image surface, blurring the image. A smaller pinhole gives us a smaller circle and a sharper image. On the other hand, the smaller pinhole results in a dimmer image since the amount of light that reaches the image surface depends on the size—specifically the area—of the pinhole. Although the situation is more complex in the

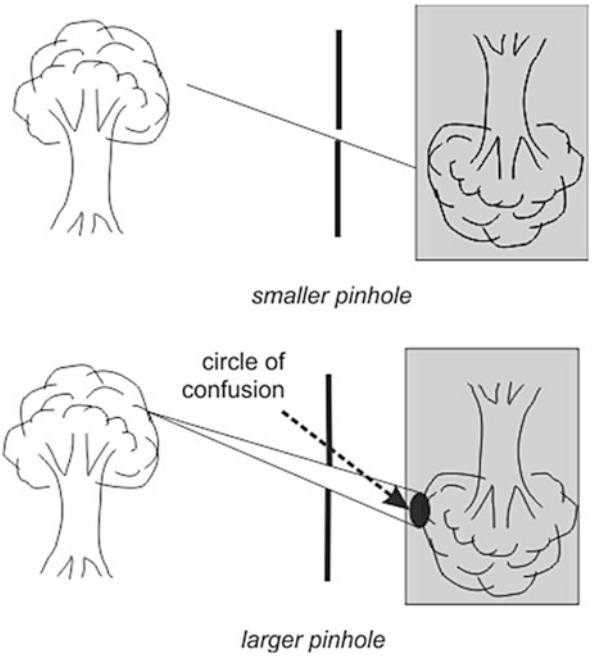


Fig. 2.2 An image formed by a pinhole

case of lenses, this trade-off between sharpness and brightness is fundamental to photography; it has practical effects on the design of cameras and even the way we take photos.

Most subjects do not generate their own light but instead reflect light from other sources. The path from light source to image surface is shown in Fig. 2.3. *Incident light* is the light that falls on a subject; light may be *direct illumination* coming directly from a source or *indirect illumination* that has been reflected from some other object. *Reflected light* is the light reflected by the subject. *Luminance* is the product of illumination onto an object and the object's reflectance. The appearance of the object—both its brightness and its color—depends on both its reflection properties and the incident light upon it. In the case of indirect illumination, the light's qualities depend not only on its original source but the objects from which it has been reflected. We will see in Sect. 2.3 that the visual system is extremely adept at adjusting for variations in lighting to maintain the consistent appearance of an object.

We use the concept of *spatial frequencies* to help us analyze images. Figure 2.4 shows a pattern of alternating light and darkness; this pattern is produced by sinusoidal variations in the luminance of the image. We can form these patterns both horizontally and vertically. We can also combine them to produce 2D spatial frequency patterns. Spatial frequencies are useful because we can compose complex patterns as combinations of basic sinusoidal spatial frequencies. We will use spatial frequencies in JPEG compression in Sect. 3.6.2.

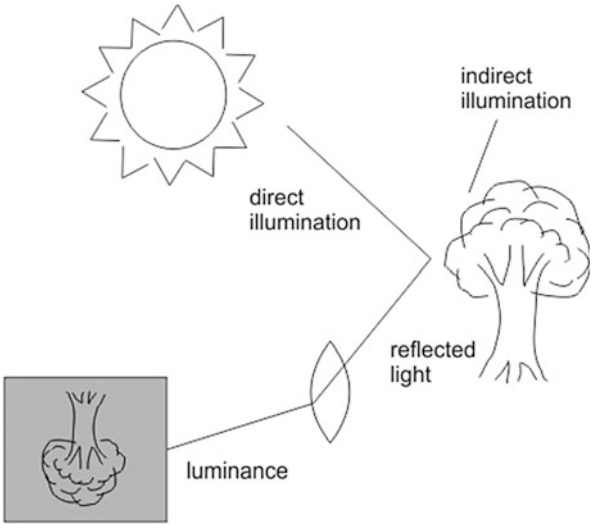


Fig. 2.3 The path from illumination to luminance

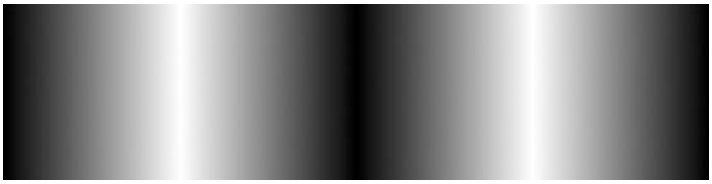


Fig. 2.4 A plot of spatial frequencies

As we will see in Sect. 2.4, color is ultimately a perceptual phenomenon that depends on how the eye and brain work. But the physical phenomenon of color is determined by the *frequency* of light. Different frequencies produce different colors in the *visible spectrum* ranging from violet to red. We often use *frequency* and *color* interchangeably.

Cameras that can take pictures in either the ultraviolet (UV) or infrared (IR) regions have their uses. Ultraviolet photography has many scientific uses. Infrared cameras are widely used in consumer cameras. However, keep in mind that these infrared cameras make use of light that is very near the visible band, known as shortwave infrared (SWIR). Subjects such as people require illumination by infrared light sources but can be captured with standard image sensors. Thermal images capture longwave infrared (LWIR), which require a fundamentally different image sensing technology. We will discuss infrared sensing in more detail in Sect. 3.4.5.

Different light sources can have very different color temperatures: incandescent lights are yellow, while fluorescent lights are green. The composition of sunlight varies throughout the day. Figure 2.5 shows that the sun at high noon goes through less atmosphere than at sunrise or sunset. As a result, sunlight is more heavily

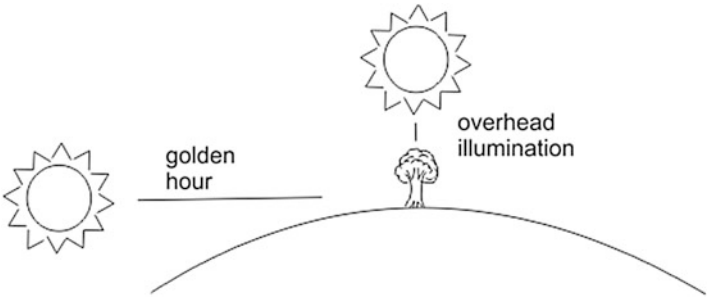


Fig. 2.5 Atmospheric filtering of sunlight and the golden hour

filtered at those times, giving it a golden hue. Photographers refer to the hour around sunrise or sunset as the *golden hour*. The human visual system perceives the orange/teal pair as having the highest contrast of any pair values on the color wheel.

2.2.2 The Physics of Light

Light is a form of electromagnetic radiation [Fey10]. Electromagnetic radiation propagates as waves that can be characterized by either frequency or wavelength. The wavelength and frequency of light are related by the speed of light c:

$$\nu = \frac{c}{\lambda} \tag{2.1}$$

We will discuss how people see in more detail in 2.3, but the term *light* is generally used for electromagnetic radiation that is at least near the range of frequencies/wavelengths that can be detected by the eye. (Radio, for example, refers to electromagnetic radiation at lower frequencies than that of light.)

Figure 2.6 shows the visible light *spectrum*, which occupies the wavelengths roughly $400 - 700$ *nm*. The wavelength of light is perceived as color. Wavelengths longer than 700 *nm* are known as infrared, while wavelengths below 400 *nm* are ultraviolet.

The human eye is not equally sensitive to all wavelengths of light. The CIE $V(\lambda)$ function, shown in Fig. 2.7, is the standard definition of the eye's relative sensitivity under typical daylight. The eye's sensitivity peaks at 555 *nm*.

We need to be able to measure the intensity of light [Per07]. Luminous intensity is a measure of power but weighted by the sensitivity of the eye. The candela is defined as light at 555 *nm* with a given power level in Watts through a given solid angle. The resulting unit is the *candela* (*cd*), roughly the illumination produced by one candle. The illuminance of a source projected on an area in a given direction is measured in *candela per square meter* (cd/m^2) or *nits*. The *lux* (*lx*) is, by comparison, a measure of intensity falling on a surface; it depends on the angle at which the

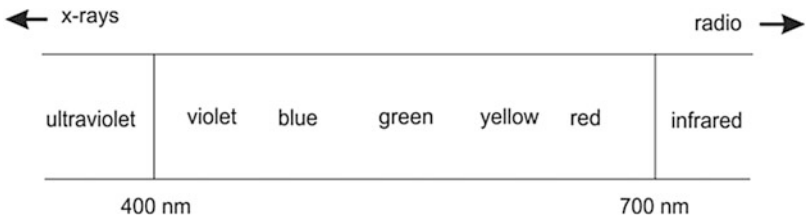


Fig. 2.6 The spectrum of light

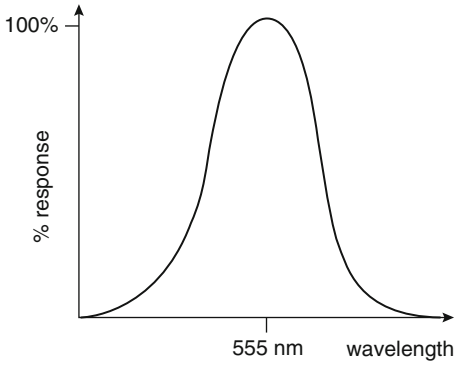


Fig. 2.7 The CIE $V(\lambda)$ function for sensitivity of the human eye [Per07]

light strikes the surface as well as the intensity and distance from the source to the surface. Illuminance is highest when the light is normal to the surface; it falls off as $\cos\theta$ as the source goes off the normal.

We use the *lumen* (*lm*) to measure the total light emitted from the source in all directions. It is derived from the candela so it also weighted by the eye's sensitivity. A 60 *W* incandescent bulb puts out about 800 *lm*.

One important physical mechanism for the generation of light is thermal. The color of light emitted by a body depends on its temperature. We use this concept to define a metric for color—*color temperature*, measured in Kelvin (K), is the temperature of a black body that emits light of that color. A black body absorbs all light falling onto it, so the only light that comes from the ideal black body is produced by its thermal radiation.

What we refer to as white light is actually a mixture of light at several different frequencies. The intensities of the various component frequencies, as well as their intensities, combine to produce the perception of the color white. We will discuss color perception in more detail in Sect. 2.3.

The French physicist Pierre de Fermat characterized the behavior of light in what has become known as *Fermat's principle:* light takes the path that requires the shortest time.

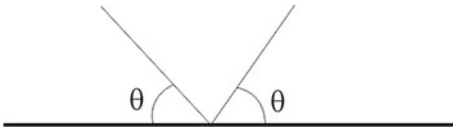


Fig. 2.8 Reflection of light

Light can be *absorbed* or *reflected* by a subject; most subjects combine absorption and reflection. Most subjects absorb light of different frequencies at different rates, which helps to determine the subject's color. Strictly speaking, reflection results from light being absorbed and then reemitted by the subject, but for purposes of ray optics, the ray appears to bounce off the reflector. For a flat ideal reflector, the angle of incidence θ equals the angle of reflection as shown in Fig. 2.8. This form of reflection is known as *specular*. Reflection can also be diffuse, in which case not all of the rays leave the surface at the same angle.

We can manipulate light in other ways by taking advantage of two additional mechanisms: *refraction* and *diffraction*. Refraction allows us to focus light. Diffraction can be useful but also acts as a nuisance factor that limits the performance of optical systems.

Refraction, illustrated in Fig. 2.9, refers to the bending of the path of light through a medium. Glass is the most common medium used to generate refraction, but high-quality optical plastics can also be used. As the boundary of the medium, the phase velocity of the wave changes, but its frequency does not. If the wavefront is perpendicular to the medium, it continues to travel through the medium in the same direction. If not, the wave's direction changes at the medium's boundary. *Snell's law* [Fey10] describes the relationship between the angles of the wavefront in the two media:

$$\frac{\sin\theta_1}{\sin\theta_2} = n \tag{2.2}$$

The refractive index n of a material is used to characterize the material. Refraction depends upon wavelength; this property results in chromatic aberrations as we will see in Sect. 2.5.

Diffraction occurs at boundaries of a different form, namely, edges. The classic case of diffraction is through a slit as shown in Fig. 2.10. In this case, we consider the light not as rays but as wavefronts. On the left side of the slit, light moves as waves that cover the entire surface. The planar wave that hit the slit becomes a cylindrical wave that moves away from the slit. The intensity of light as a function of position is highest in front of the slit, but several smaller peaks are also formed. The angle between the two minima closest to the maximum intensity is given by the *Fraunhofer diffraction formula*:

$$\alpha \approx \frac{2\lambda}{W} \tag{2.3}$$

The more general model for diffraction is *Kirchhoff's diffraction formula*.

Fig. 2.9 Refraction of light

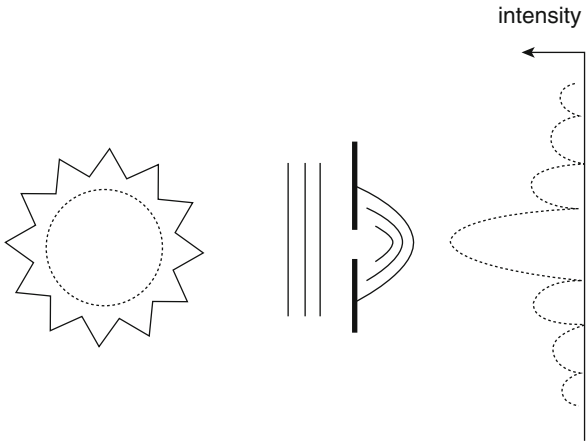


Fig. 2.10 Diffraction of light

When light passes through two nearby slits, the two waves emanating from the slits *interfere* with each other. The intensity at any point is the sum of the intensities of the two waves. The intensity of this interference pattern is the superposition of the intensities due to each slit. A regular pattern of slits, illustrated in Fig. 2.11, is known as a *diffraction grating*. These two concepts are together known as the *Huygens-Fresnel* principle. In the case of light through a circular aperture rather than a slit, the intensity pattern is known as the *Airy diffraction pattern*.

Polarization is a physical phenomenon that has practical uses in the formation of useful images. Light waves can be oriented at different angles. Glare from shiny surfaces consists of a lot of light oriented at many different angles. A polarizing filter selects light waves only at certain orientation, blocking the rest. Polarizing filters can be used to eliminate glare, which is often more strongly oriented than is light from other parts of the image. A polarizing filter can also be used to darken blue skies in color photographs. (A red filter can be used to darken the sky in a monochrome image.)

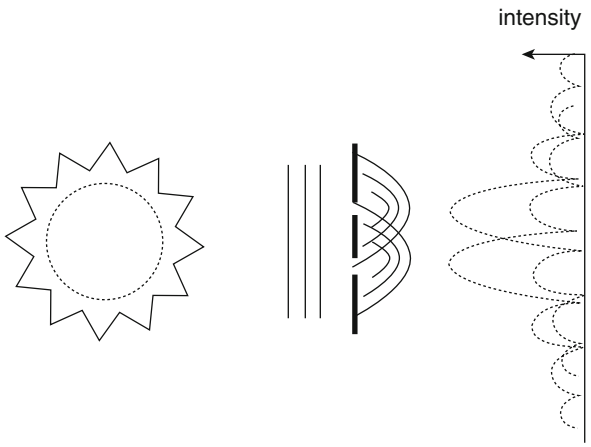


Fig. 2.11 A diffraction grating

2.3 The Human Visual System

Our eyes function as cameras, but only in part. We see the world through a complex visual processing system that starts within the eye and stretches to a large part of the brain. If we want to use cameras to make images on paper or screens that give us a sense of what the world looks like, we have to understand vision and how it differs from a simple camera. Our discussion here only scratches the surface of a subject that is both very complex and still under development. In addition to wondering at the complexity of visual processing, we can also identify some perceptual mechanisms that will help us understand how to manage exposure and color in cameras.

Figure 2.12 gives a highly simplified view of some of the components of the visual system [Pal99]. The eyes connect to the optic nerves. The nerves from the two eyes connect at the optic chasm. Several other structures—the lateral geniculate and superior colliculus—feed into the optic radiations, which then connect to the visual cortex. At each stage in this system, the information gathered from the eyes is processed and manipulated.

The eye does not have a shutter. The response of the retina to illumination at any given instant decays over a short period due to the electrochemical processes that transform light into neural pulses.

Figure 2.13 shows the structure of the eye. The top diagram shows the major structural elements. Incoming light is mediated by the cornea, aqueous humor, iris, and lens. The lens is flexible; the ciliary muscles stretch and relax the lens to change its shape and focus. Muscles also control the iris to determine the amount of light coming into the eye. The eye is not empty but filled with vitreous humor. The retina covers much of the inner surface of the eye. The bottom diagram shows the structure of the retina in more detail. The fovea is the optical center, covering only about two degrees of the visual field. The retina has two types of photoreceptors: cones are adapted to color and fine detail and rods are more sensitive to light and used primarily for low light levels. The area outside the fovea has a few cones,

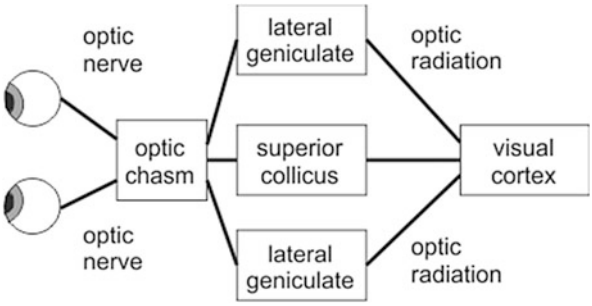


Fig. 2.12 Major components of the visual system

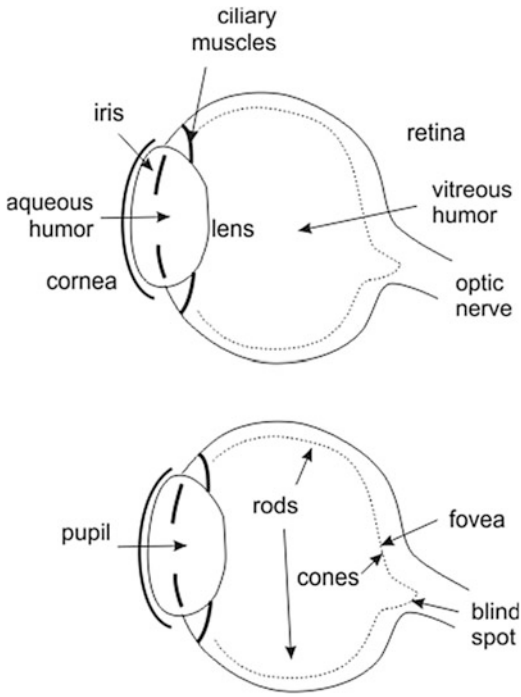


Fig. 2.13 Structure of the eye

but most of the retina is occupied by cones. The fovea is much more densely covered with photoreceptors than is the rest of the retina. Nerves from the retina gather to send data to the optic nerve; that point is a small blind spot.

Visual acuity refers to the resolution of the visual system. The visual system can resolve two lines about 1 arc minute apart, primarily due to the cones in the fovea. Visual acuity is limited primarily by the lens.

The eye uses a lens to throw an image onto the retina, but that is about the limit of the comparison to a camera. The visual system processes visual information in many different ways:

- The eye resolves only a small part of the scene at high resolution at any given time. The eye is constantly in motion, painting different parts of the scene onto the fovea. The visual system builds up our perception of the scene from this scan.
- The retina uses a lateral inhibition network of neurons to increase the contrast of the scene. Each photoreceptor is connected to nearby photoreceptors. When the photoreceptor is exposed to high illuminations, it fires rapidly to indicate the illumination level. The lateral inhibition network transmits these high firing rates to the nearby photoreceptors and inhibits their activity. The inhibited photoreceptors fire at lower rates than would be expected otherwise, increasing the difference in perceived illumination between the two sites. The result is higher contrast.
- The optic chiasm reorganizes the neural pathways from the eyes. Before the chiasm, the optic fibers transmit information entirely from one retina. After the optic chiasm, the signals from both eyes corresponding from the left half of the visual field go to the right, while signals for the right half of the visual field go to the left.
- The superior colliculus helps to control eye movement.
- The lateral geniculate nucleus controls the vergence and focus of the eyes and analyzes the position of major objects.
- The visual cortex performs a number of analytic functions. The visual field is mapped onto the visual cortex; depth in the visual cortex roughly corresponds to the complexity of the objects being analyzed. For example, early layers identify lines at each point in the visual field, with lines identified at many different orientations; later layers combine these line segments into curves.

Even before we worry about how we perceive objects, we need to understand some basic visual mechanisms that are directly relevant to photography. How do we perceive brightness? And how do we perceive color?

The visual system's response to light is not proportional. The visual system can respond to huge variations in light levels; proportional response is difficult over such large ranges of values. Over moderate luminance levels, the visual system responds logarithmically, known as the *Weber-Fechner law*; over wider ranges, a power law model is more accurate. We will use this relationship between stimulus and response to understand exposure in Sect. 2.8.

Illumination levels vary widely both within and between scenes. Evening and full noon sun provide very different levels of light; our perception of illumination between these two scenes is smaller than the physical difference. As we look around a scene, objects can be at very different reflectances. Adams [Ada02B] observed that natural scenes can easily show ratios of reflectance between the brightest and least bright object of 200-to-1. *Constancy* refers to our perception of object properties such as reflectance and color independent of variations in illumination, angle of view, etc.

Visual scanning helps the visual system to manage varying lighting levels and maintain lightness constancy. As the eye moves around the scene, the iris size is adjusted to control the retina's exposure levels. But this dynamic adjustment

mechanism is not sufficient to explain constancy—some other system must control the iris. The ratio of luminances at an edge provides cues for the determination of perceived lightness. The visual system also needs to determine absolute levels. The visual system seems to use an *anchoring heuristic* that assigns white and black at the extreme ends of the luminances in the scene [Pal99]. Anchoring requires the visual system to perform some global analysis of the scene. We will compare this characteristic of the visual system to our selection of exposures for images in Sect. 4.3.

The eye does not scan randomly. *Saliency* refers to the characteristics of a scene that tend to draw attention by the visual system. Generally speaking, edges and detail are salient. We will return to computational models of saliency in Sect. 5.3.1.

Our *color constancy* allows us to see a red ball as red even when viewed under different color conditions. Sunlight and indoor fluorescent lighting, for example, have very different spectral compositions. As a result, the light reflected off the ball will have very different spectral characteristics in each situation. But we still see the ball as red in both situations. Land[1] and McCann [Lan71] proposed the *retinex theory* to help explain this phenomenon. They used pictures they called *Mondrians* to study this problem; each image consisted of rectangular patches of varying sizes, colors, and values. They proposed that the visual system finds the ratio of reflectances on the two sides of an edge, then chains together these ratios from one region to the next to be able to compare reflectances between objects in different parts of the visual field. Retinex theory also combines local, edge-related measures with global information. It can be used to adjust for variations in color temperature; it can also handle slow variation of lighting within the scene. We will consider the problem of adjusting photographs for illumination color temperature in Sect. 3.5.2.

The capture and presentation of motion in video is based upon *apparent motion*. Video is composed of a set of still images. The *real motion* of an object, in contrast, is continuous. Modern cinema is presented at 24 frames per second; traditional video is presented at 30 frames/sec. Video takes advantage of the *beta effect* to present the illusion of motion from image sequences [Pal99]. This effect results in the seemingly continuous motion of a light that alternates between two positions at about 10 frames per second. (The *persistence of vision* theory as an explanation of the effects of motion pictures has been disproven.) These frame rates are not, however, fast enough to prevent the perception of flicker. At 60 frames per second, *flicker fusion* results in the perception of continuous illumination. As we will see in Sect. 2.disp, many video display systems use different techniques to increase the display rate to the flicker fusion rate without increasing the actual frame rate.

[1] Edwin Land also invented the process used to manufacture polarized optical material as well as the Polaroid instant photography process.

2.4 Color Science

Color is a critical aspect of visual perception. The visual mechanisms of color are sufficiently important that they deserve separate consideration. We will first study theories of color vision. We will then move onto models of color that bridge the gap between perception and reproduction.

2.4.1 Theories of Color Vision

Color science combines physiology, engineering, and art. In order to accurately and predictably capture and reproduce color, we first need to understand a little more about how the visual system perceives color. The retinex theory from Sect. 2.3 describes a later step in the process; early stages determine what colors we can see.

The *tristimulus theory* describes the way that the retina responds to light. Three types of cones respond to three different bands of light. Figure 2.14 shows their relative response to frequencies of light; the green-responsive cones are most sensitive and provide the largest absolute response. Together, these three receptors cover the visible light range. The eye does not respond to all wavelengths of light equally well—it is most sensitive to green, a fact that we will exploit for image sensors in Sect. 3.sensor. Our sensation of color starts with the relative amount of stimulation of the three types of cones at a given location.

This leads to the *additive color* system that you probably learned as a child. As shown in Fig. 2.15, the three *primary colors* are red, green, and blue. We combine these primary colors to create other colors. The system is called *additive* because we add together the primaries as if we are shining primary-colored lights onto a white surface that reflects all colors. However, printed material behaves as a *subtractive* medium—white light shining on a patch of a given color will absorb some of the wavelengths and reflect others. The *subtractive color* system is based on the *secondary colors* yellow, cyan, and magenta. The primary system is referred to as *RGB*. The secondary system is typically referred to as *CYMK*—printing processes generally require a separate black to ensure saturated dark areas, which is labeled *K*.

However, the visual system cannot in fact distinguish all types of color combinations. One part of the visual system makes use of *opponent process theory* to reduce the amount of information that later stages need to process [Pal99]. This encoding describes color as pairs of opposites: red/green or blue/yellow. As a result, there are no colors that subjectively combine red/green or blue/yellow.

The *perceived* color of an object depends on three components:

- The color of the light with which the object is illuminated
- The reflectance of the object
- The response of the visual system to the reflected light

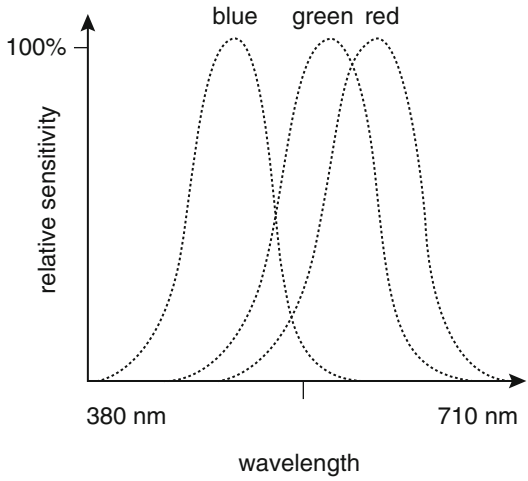


Fig. 2.14 Response characteristics of cones in the retina

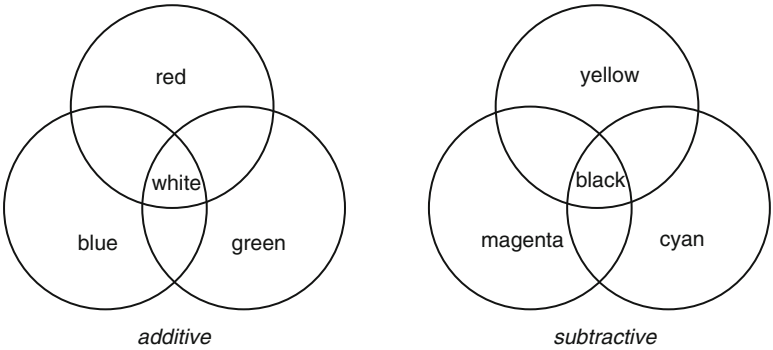


Fig. 2.15 Additive and subtractive color systems

The visual system is very well adapted to correct for variations in illumination. However, cameras are not. We will see the effects of illumination on images in Sect. 2.8.1.

2.4.2 Color Models

Given the complexities of how we perceive color, we need ways to define color beyond the basic additive/subtractive approach. An early color dictionary was created by Albert Munsell. The *Munsell Book of Color* consists of 1600 carefully manufactured paint chips that were designed to capture just noticeable differences in color. The book is still widely used to identify colors of objects by comparing its color to samples in the book.

The Commission International de l'Eclarage (CIE) established the first mathematical color specification in 1931 [Per07]. A color sample is described by XYZ tristimulus values. The XYZ value is the result of the combination of the spectral radiance of a light source $S(\lambda)$, the spectral reflectance $R(\lambda)$, and a standard observer model $\bar{x}(\lambda), \overline{y}(\lambda), \bar{z}(\lambda)$:

$$X = k \int S(\lambda)R(\lambda)\bar{x}(\lambda)d\lambda \tag{2.4}$$

$$Y = k \int S(\lambda)R(\lambda)y(\lambda)d\lambda \tag{2.5}$$

$$Z = k \int S(\lambda)R(\lambda)\bar{z}(\lambda)d\lambda \tag{2.6}$$

where k serves as a normalizing factor to ensure a maximum value of 100. The $\overline{y}(\lambda)$ is the CIE $V(\lambda)$ intensity function shown in Fig. 2.16. $\bar{x}(\lambda)$ and $\bar{z}(\lambda)$ take the form shown in Fig. 2.16. This diagram is known as a *chromaticity diagram* because it eliminates the value dimension, using only hue and saturation.

This model was created before a much experimental data on the range of perceived colors was available. Over several decades, it became clear that the CIE color space did not match well to a *just-noticeable-difference* (*JND*) model of observable color—the distance between two colors in the diagram did not correspond well to the perceived difference between the colors. More uniform color spaces have been proposed: CIELAB, ΔE^*94, and CIEDE2000. However, the CIE chromaticity diagram is still widely used to explain color spaces.

The HSV color space is a widely used color space. We can describe color using three criteria, each forming an axis of the color space:

- *Hue* is what we colloquially call color. It corresponds to the dominant wavelength of light reflected by the color.
- *Saturation* refers to the color's purity.
- *Lightness* or *value* refers to the color's relationship to the range between black and white.

Figure 2.17 shows the color spindle model for HSV. Hues vary around the circumference of the spindle. Saturation varies from a completely unsaturated gray in the middle to fully saturated at the edges. Value varies from black to white along the spindle's axis.

The *YUV* color model comes from the US analog color television standard, which was created to be compatible with existing monochrome broadcast standards. The Y component describes luminance, while U and V describe chrominance. The terminology *YCrCb* is used for the digital versions of this approach to representing color. The relationship between YUV and RGB is defined by the analog broadcast standard and can be described by a matrix:

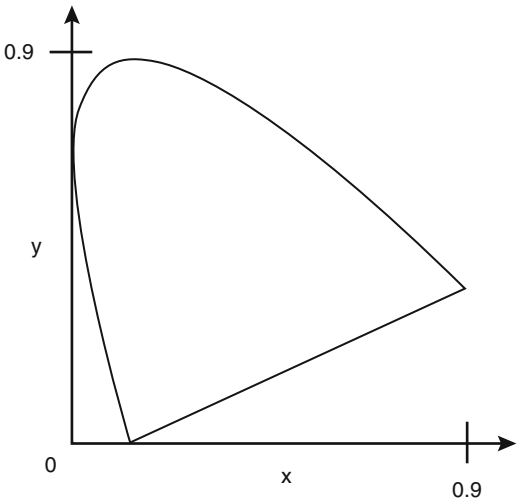


Fig. 2.16 The CIE chromaticity diagram

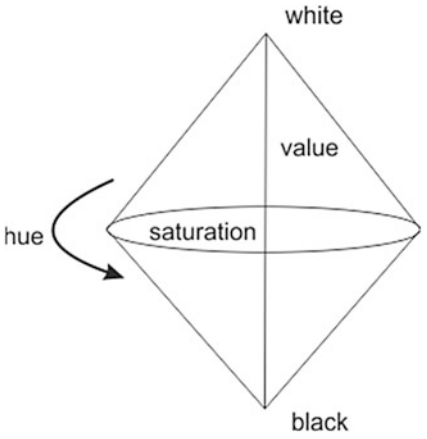


Fig. 2.17 The HSV color spindle

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.13983 \\ 1 & -0.39465 & -0.58060 \\ 1 & 2.03211 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{2.7}$$

2.5 Lenses

As we saw in Sect. 2.2, forming images requires controlling light. Lenses give us much greater control over light than do pinholes. As with any design problem, we must make trade-offs when designing a lens.

2.5.1 Lenses and Image Formation

We can visualize the behavior of a lens using ray tracing. Figure 2.18 shows a simple lens and the path of two rays through the lens. At each lens surface, the ray refracts or changes angle. Two rays are enough to help us understand basic effects of the lens.

Focus is a fundamental property of an image. As with many aspects of photography, perceptual and artistic properties play a role, but we can understand the basic physical principle. As shown in Fig. 2.19, parallel beams coming into the lens along the lens' axis converge at the *focal point*. Beams from different directions will be focused to form a *focal plane*. Given a subject at infinity, the distance from the lens to the focal plane is known as the *focal length*. Parallel or collimated beams correspond to light from a subject that is infinitely far away. Even though the focal plane distance changes with distance to the subject, we still use the focal distance at infinity as a basic characteristic of the lens. The lens' focal length is commonly referred to as *f*.

Subjects at varying distances from the lens will focus at different distances from the lens. The lens creates an image volume which we can sample at different points to find different parts of the image in focus. The eye changes focus by pulling or relaxing the lens to change its curvature. Flexible lenses are the rare exception for cameras. Instead, the lens is moved mechanically relative to the image surface to align the image surface with the desired focal plane.

Image planes near the focal plane are nearly but not quite in focus. Light in those planes forms not a point but a *circle of confusion* as shown in Fig. 2.20. We are willing to live with a small circle of confusion for two reasons. First, our eye is limited in its resolving power. Second, most objects are not perfectly flat but have depth; different parts of the object will focus at different planes, and there is no single plane in which an entire 3D object will be in perfect focus. Eastman Kodak gives 0.05 *mm* as the size of a just-acceptable circle of confusion for a standard 24×36 *mm* image surface [Kod88]; larger image surfaces allow for larger just-acceptable circles of confusion. We refer to the range of planes at the subject that come into acceptable focus as the *depth-of-field*. Limited depth-of-field is a drawback in some images and an advantage in others. Deep depth-of-field can give a sense of realism in landscapes; we can use limited depth-of-field to draw attention

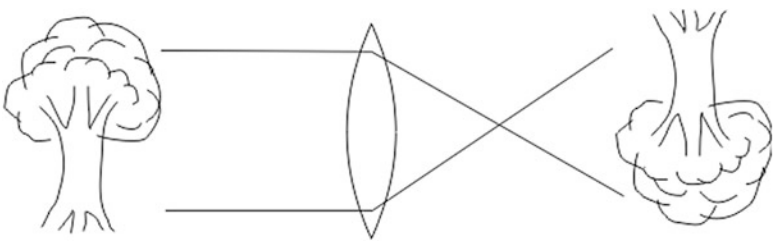


Fig. 2.18 Ray tracing a lens

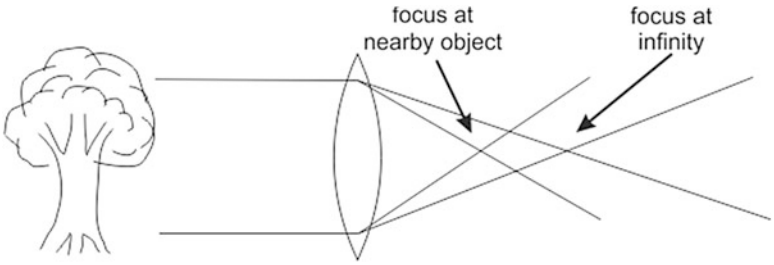


Fig. 2.19 Image focus

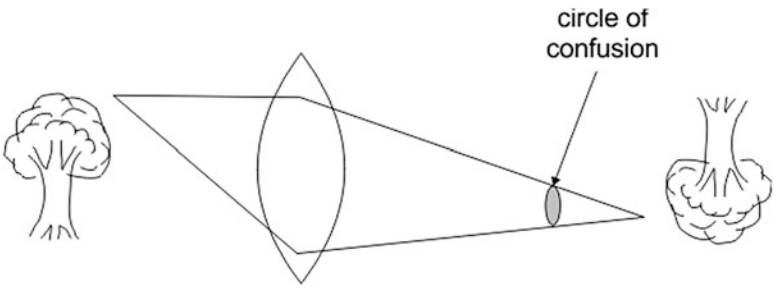


Fig. 2.20 Circles of confusion

away from the out-of-focus areas to subjects that are in focus. In practice, we use the focal length to characterize the angle of scene covered by the lens.

Bokeh is a term for the rendering of highly defocused elements, typically in the background. Bokeh is influenced by lens aberrations; the shape of the aperture also plays a role. Portraits are often shot with small depth-of-field, and bokeh of background elements is considered an artistic element. Some lenses use a radially oriented graduated neutral density filter to shape bokeh [San17]; the filter rounds off the edges of the bokeh elements, which is particularly helpful for highlights and flare.

Lens focal length determines the size of the image circle thrown. As shown in Fig. 2.21, shorter focal length lenses throw a smaller image circle, while longer focal lengths throw larger image circles. If we do not change the size of the image surface we use to view the image, longer focal lengths give us a narrower view of the subject since the image surface covers a smaller proportion of the image circle. Short focal lengths squeeze the subject onto a smaller image circle, more of which fits onto the image surface. We choose short focal length lenses (or simply *short* lenses) to give a wide view of the scene; we use long lenses to pick a small part of the scene.

Conversely, changing the image surface size changes the image coverage as shown in Fig. 2.22. Different types of camera use different image surface sizes: cell phones use small image sensors, while dedicated cameras generally use image sensors that are considerably larger. As we move to larger image surfaces, the

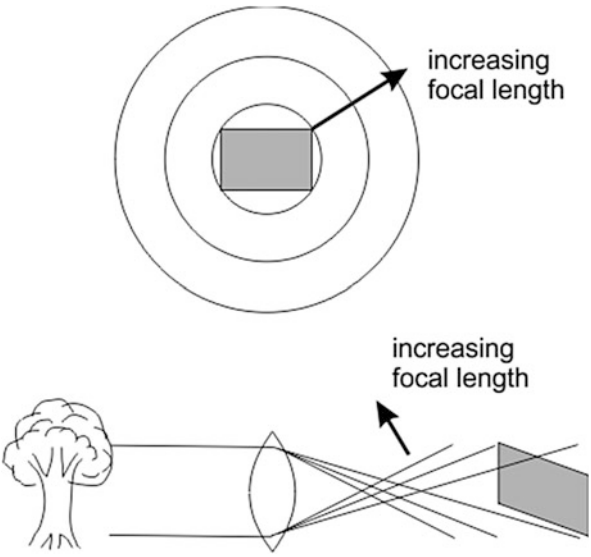


Fig. 2.21 Focal length determines image circle size

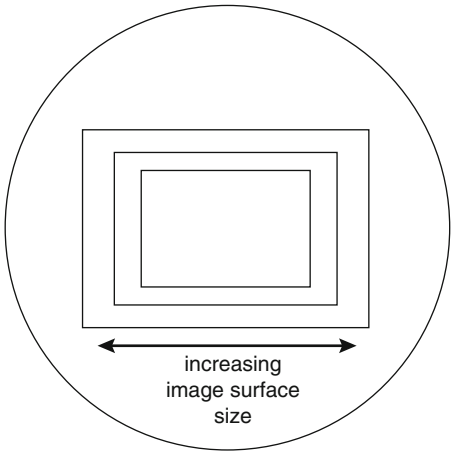


Fig. 2.22 Focal length and image surface size

image surface covers more of the scene and the lens appears to be less selective. The lens has not changed, only our use of the image it throws.

A *normal* lens is one that gives a field-of-view that is equal to that of the human eye. This occurs when the focal length of the lens is about equal to the diagonal size of the image surface. As a result, the definition of a normal lens depends on the image surface size. 50 *mm* is a normal lens for the standard 35 *mm* full-frame format, which uses an image surface of 24 *mm* $\times$ 36 *mm*. A cellphone with a small image sensor has a shorter normal focal length; a larger image sensor would require a longer lens to achieve a normal field-of-view.

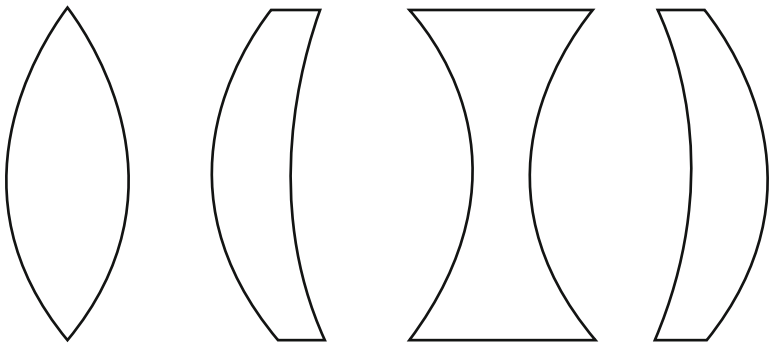


Fig. 2.23 Lens shapes

2.5.2 Ray Optics

The optics of photographic lenses is relatively simple by the standards of modern physics. Nonetheless, the design of modern photographic lenses is a complex problem with elements of science, engineering, and art.

Spherical lenses are simplest to analyze. They are also simplest to manufacture. Many lenses are manufactured today using a centuries-old technique: a block of glass is ground to shape using a form with the inverse of the lens' shape with the aid of an abrasive.

As shown in Fig. 2.23, each side of the lens can be concave, convex, or flat. The shapes of the fronts and backs of the lenses can be combined giving eight basic configurations. Each side is characterized by its radius of curvature r. Lenses may also vary considerably in their thickness.

The *lensmaker's formula* gives the focal length f of a thin lens [Fow75]:

$$\frac{1}{f} = (n-1)\left[\frac{1}{r_1} - \frac{1}{r_2}\right] \tag{2.8}$$

where n is the refractive index of the lens, d is the lens thickness, r_1 is the radius of curvature of the lens side closest to the light source, and r_2 is the radius of curvature of the side of the lens away from the light source. *Diopter* is the reciprocal of focal length.

When a series of lenses is placed in contact, their combined focal length is

$$\frac{1}{f} = \frac{1}{f_1} + \frac{1}{f_2} + \cdots \tag{2.9}$$

When two lenses are separated by distance d, their combined focal length is

$$\frac{1}{f} = \frac{1}{f_1} + \frac{1}{f_2} - \frac{d}{f_1 f_2} \tag{2.10}$$

When a subject is not at infinity but at a finite distance from the lens, the focal plane is at a different position than for the infinity case. The lens focuses rays from a point on the subject to a single point, allowing an image to form. However, the focused image will not be at the focal length. The relationship between the subject-lens distance and the lens-image plane distance is

$$\frac{1}{s_1} + \frac{1}{s_2} = \frac{1}{f} \tag{2.11}$$

where s_1 and s_2 are the subject/lens and lens/image plane distances, respectively.

The *nodal point* of a lens is the point through which rays travel such that the entry angle of the ray is the same as its exit angle. A lens has two nodal points, forward and backward.

We can determine the width of the region for which the image is in acceptable sharpness. We need a criterion to determine acceptability; we use the circle of confusion diameter c as the specification of sharpness. The *hyperfocal distance* is the distance for a region of acceptable sharpness that extends to infinity—any object farther away than the hyperfocal distance will be in focus:

$$H = \frac{f^2}{Nc} + f \tag{2.12}$$

where N is the f-stop to which the lens is set and c is the circle of confusion diameter. For some other subject distance s, the region of acceptable sharpness is in the range

$$\left[\frac{s(H-f)}{H+s-2f}, \frac{s(H-f)}{H-s}\right] \tag{2.13}$$

The iris, shown in Fig. 2.24, is built into the lens and can be adjusted to create a larger or smaller hole and thus controlling the amount of light reaching the image surface.

A virtual image is formed on the same side of the lens as the image. As shown in Fig. 2.25, the rays through the concave lens rays diverge on the imaging surface side but converge on the subject side. The image is virtual because rays do not travel through the space in which the image appears to be. A magnifying lens is a practical example of the use of virtual images—the virtual image formed by the lens appears larger than the subject. Mirrors also throw virtual images.

Fig. 2.24 An iris in a lens

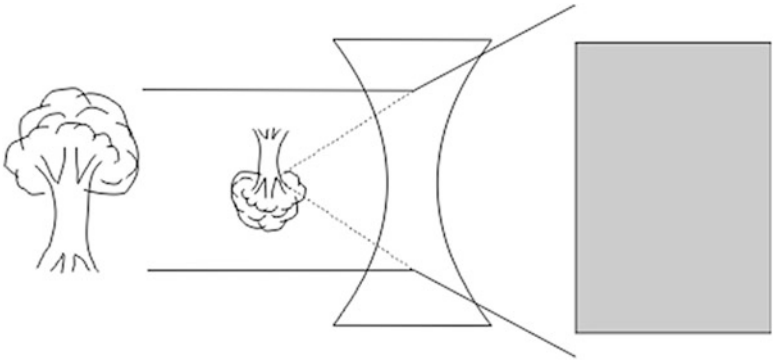


Fig. 2.25 Formation of a virtual image

2.5.3 Lens Design

Although the basic physics of lenses has been understood for centuries, lens design presents a substantial engineering problem. The creation of modern lenses requires careful engineering to provide a balanced set of characteristics in the face of many competing concerns.

A *cine lens* provides several features that make the lens more suitable to the demands of long takes. Cine lenses are generally marked in T-stops, which measure the total light through the lens, rather than f-stops. Cine lenses are often designed with wide apertures to allow for shallow depth-of-field that helps to separate the

subject from the background. Zoom cine lenses are generally parfocal so that the focus does not change during a zoom. Cine lenses may also be designed as sets with common optical and mechanical properties for all the members of the set.

We build compound lenses to both add features and reduce *aberrations*. High-performance lenses combine lens elements of multiple shapes and several different optical materials.

An *aberration* is any unwanted optical behavior of a lens. These aberrations can be analyzed and characterized. Aberrations are generally categorized into six different types [Cox66]:

- Spherical aberrations
- Coma
- Astigmatism
- Oblique spherical aberrations
- Distortion
- Chromatic aberrations

As shown in Fig. 2.26, a lens that is not truly spherical will cause parallel rays at different positions to cross the lens axis at different positions.

Coma occurs for off-axis images and is a result of a finite aperture. As shown in Fig. 2.27, parallel off-axis rays will not all focus at the same position on the image plane. They will instead form a cone shape caused by circles of different sizes being formed at offsets.

Astigmatism is the result of the image being projected to a curved surface rather than a flat plane. Figure 2.28 shows a sample target which can demonstrate two effects: circles around the image center can become increasingly out of focus with distance from the center; or lines radiating from the image center can become increasingly out of focus with distance from the center. Astigmatism results in *field curvature*—the focal region of the lens is not flat.

Oblique spherical aberration results in flare surrounding the astigmatic lines, with increasing flare at greater distance from the center.

Distortion results in changes to the relationships between lines. Figure 2.29 shows two types of distortion: barrel distortion pushes the edges of a square outward and pincushion distortion pushes the edges inward.

Chromatic aberrations result from dispersion—different wavelengths of light are refracted by different amounts. As a result, they focus at different points. Lateral chromatic aberrations cause different wavelengths to focus at different points on the image plane; longitudinal chromatic aberrations cause different wavelengths to focus at different image planes.

The Cooke triplet [Cox66], shown in Fig. 2.30, illustrates how compound lenses can improve image quality; this design was a significant advance in optics for its combination of low aberrations and simple design. *Petzval field curvature* is an aberration in which a flat object produces a curved focused image. The *Petzval sum* for a set of lenses is

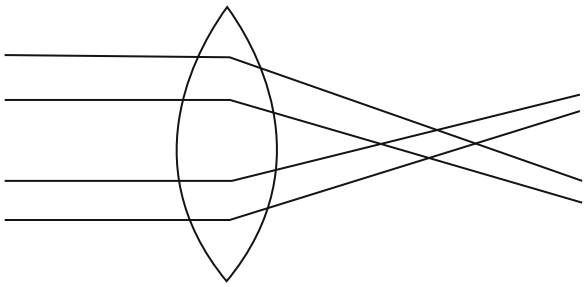


Fig. 2.26 Spherical aberration

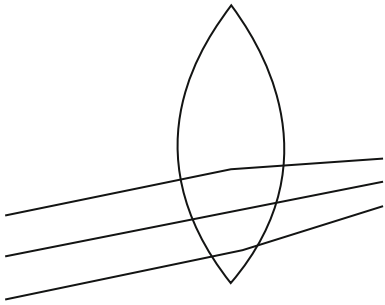


Fig. 2.27 Coma

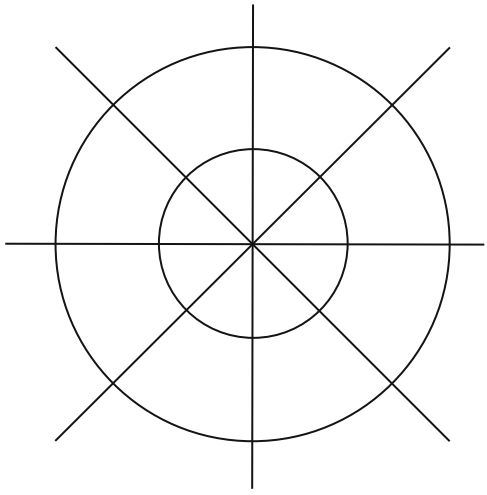


Fig. 2.28 Astigmatism

$$\sum_{1 \le i \le n-1} \frac{n_{i+1} - n_i}{r_i n_{i+1} n_i} \tag{2.14}$$

where r is the lens element radius and n is its index of refraction. The curvatures and lens materials for the Cooke triplet are selected so that its Petzval sum is 1, resulting in a flat field of focus.

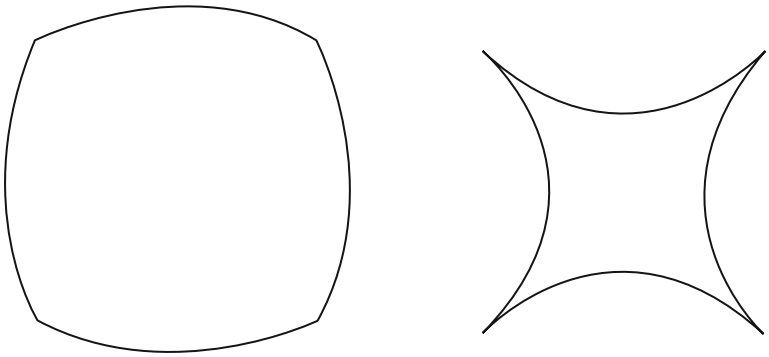
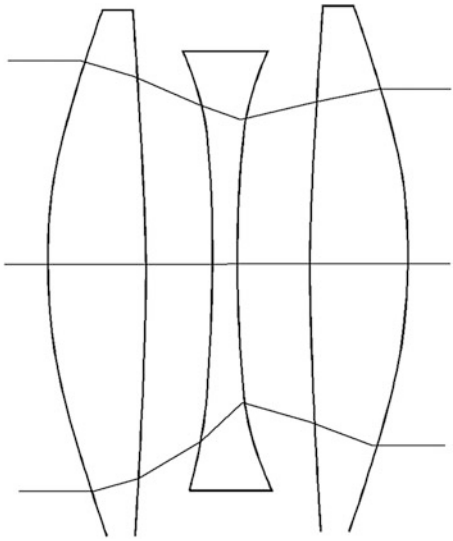


Fig. 2.29 Barrel and pincushion distortion

Fig. 2.30 The Cooke triplet



We can control chromatic aberrations over a small range of wavelengths by combining two lenses, made of different materials, each with a different dispersion [Fow75]. This structure is known as an *achromatic* lens. If the relative dispersions of the two lens elements are

$$\delta_1 = \frac{1}{n_1 - 1}\frac{dn_1}{d\lambda}, \delta_2 = \frac{1}{n_2 - 1}\frac{dn_2}{d\lambda}, \tag{2.15}$$

then the focal lengths of the lens elements are

Fig. 2.31 An example of lens flare

$$f_1 = f\left(1 - \frac{\delta_1}{\delta_2}\right), f_2 = f\left(1 - \frac{\delta_2}{\delta_1}\right) \tag{2.16}$$

The advent of high-performance optical plastics has allowed lens designers to make much more extensive use of aspheric lens elements. These aspheric elements can provide substantial corrections with fewer lens elements than would be required for purely spherical elements.

Although not an aberration, *flare* is an unwanted property of images. Flare displays on the image surface due to internal reflections off the surfaces of the lens elements. Modern lenses are *coated* to reduce flare. However, flare can still occur in some situations, particularly when a strong light source is in the image; Fig. 2.31 shows an example of flare from the sun. We can also minimize flare using a *lens hood* to protect the lens from light entering at large angles.

We often use the term *telephoto* generically to mean a long focal length lens. The technical definition is a lens whose back focus point is substantially shorter than the lens' focal length. Telephoto design allows the lens to be more compact. Telephoto lenses make use of a positive group of lenses followed by a negative group [Cox66].

A *zoom* lens can be adjusted to provide different focal lengths; the focal plane does not move as the lens is zoomed. If this were not the case, not only would the image need to be refocused after a zoom, but the zoom itself would become more difficult as the image blurred. We refer to a non-zoom, fixed focal length lens as a *prime*. Figure 2.32 shows a simple zoom lens configuration [Cox66]: the two negative lens elements move together relative to a pair of positive lenses; a prime

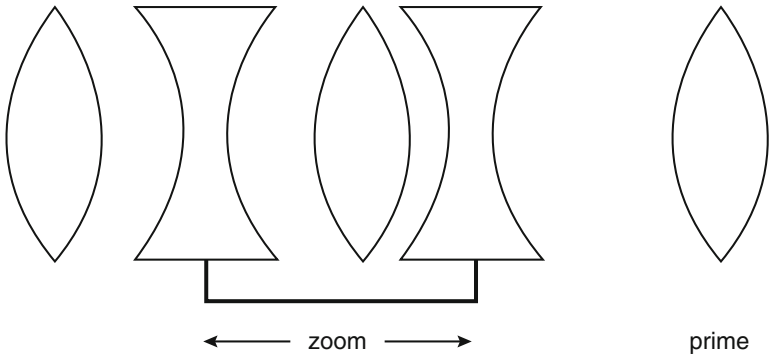


Fig. 2.32 A simple zoom lens configuration

lens forms the back of the zoom. Practical zoom lenses are much more complex to provide adequate aberration correction over the zoom range.

A substantial exception to the rule of spherical lens elements is the *anamorphic* lens, widely used in cinema to create wide-screen presentations. Figure 2.33 shows one type of anamorphic lens using cylindrical lenses. The cylindrical element curves along the horizontal image axis but not the vertical axis. As a result, it squeezes the image horizontally but not vertically. Ray tracing works in both directions, so by playing back the image using the same type of anamorphic element, we remove the distortion and produce a wider image. Anamorphic lenses are used in film to provide a wide-screen image without requiring image frames with very long, thin aspect ratios.

Some imaging characteristics rely on the relationship between the focal length and image surface size, while others depend on the absolute characteristics of the lens. For example, depth-of-field depends on absolute aperture size, not f-stop. As a result, the normal lens for a smaller image sensor gives a larger depth-of-field than does a normal lens for a larger image sensor.

2.5.4 Panoramas

Panoramic images—images which capture a long horizontal view of a scene—are almost as old as photography. Panoramic images were widely created in the nineteenth century; both the Library of Congress and Denver Public Library have collections of historic panoramas. We can create panoramas either by algorithmically stitching together several small photos or by optical means. We concentrate here on optical panoramas; we will discuss stitching algorithms in Sect. 4.mosaic.

The simplest approach, shown in Fig. 2.34, is to crop the image to a wide aspect ratio. The lens must produce an image circle large enough to accommodate the panoramic field. Much of the image circle is wasted.

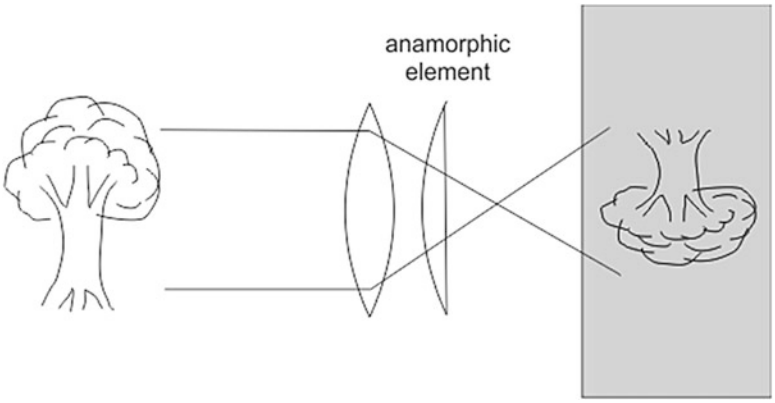


Fig. 2.33 An anamorphic lens system

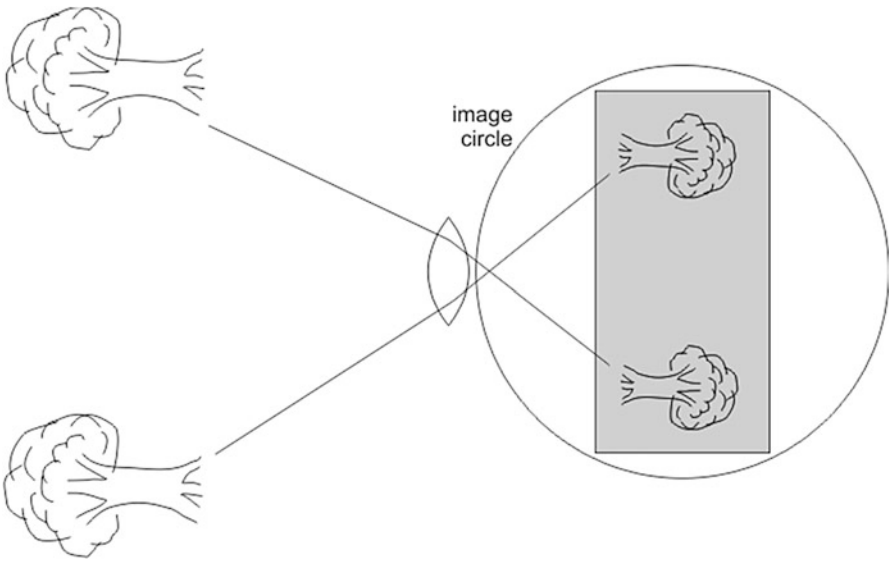


Fig. 2.34 A cropped panorama

A slightly more sophisticated approach is shown in Fig. 2.35. In this case, the lens is rotated horizontally to scan a smaller image circle across the image surface. The image surface itself is curved to maintain the focal distance to the lens. A traveling slit allows only a small part of the image to hit the image surface at each point. We minimize perspective shifts by rotating the lens around its nodal point.

Figure 2.36 shows an improved version of this method. The lens, a slit, and the image surface all rotate together. The scene is painted as a cylinder as it is scanned. The diameter of the cylindrical image depends on the focal length of the lens—longer focal lengths produce larger diameter panoramas. Figure 2.37 shows two

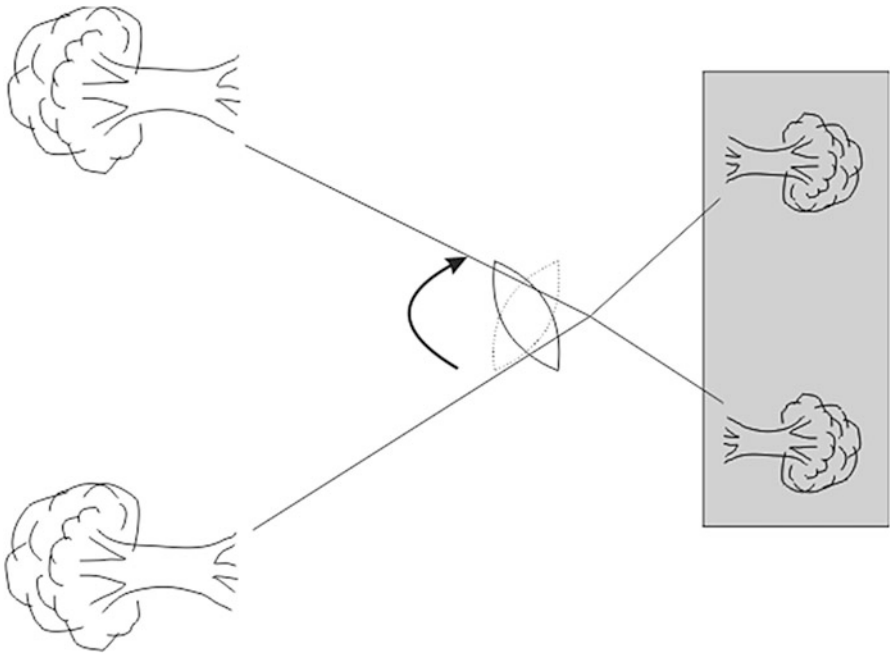


Fig. 2.35 A scanned panorama

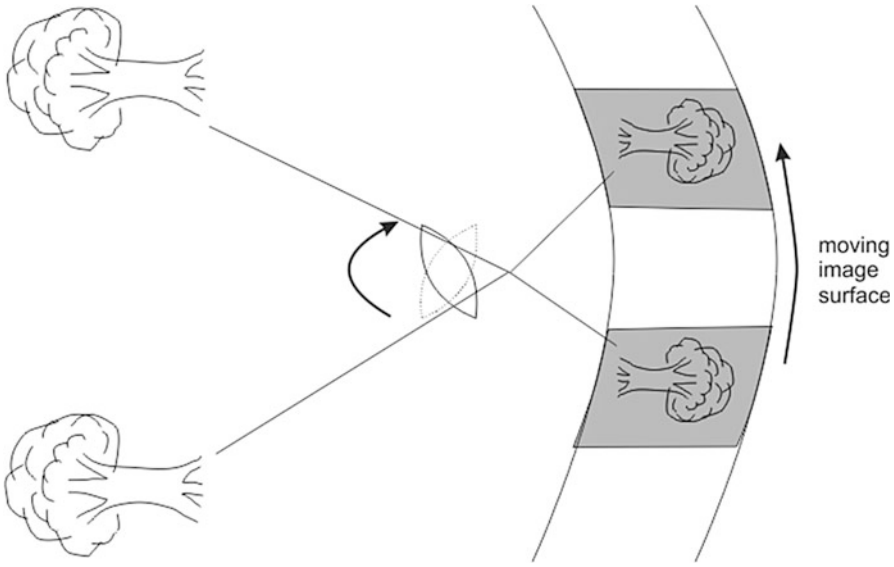


Fig. 2.36 A cylindrical panorama

short focal length

long focal length

Fig. 2.37 Panoramas taken at different focal lengths

panoramas, one taken with a longer focal length and the other with a shorter focal length. In this case, the panorama resulting from the shorter focal length diminishes the grandeur of the scene; in other cases, the smaller-diameter panorama may be the appropriate aesthetic choice.

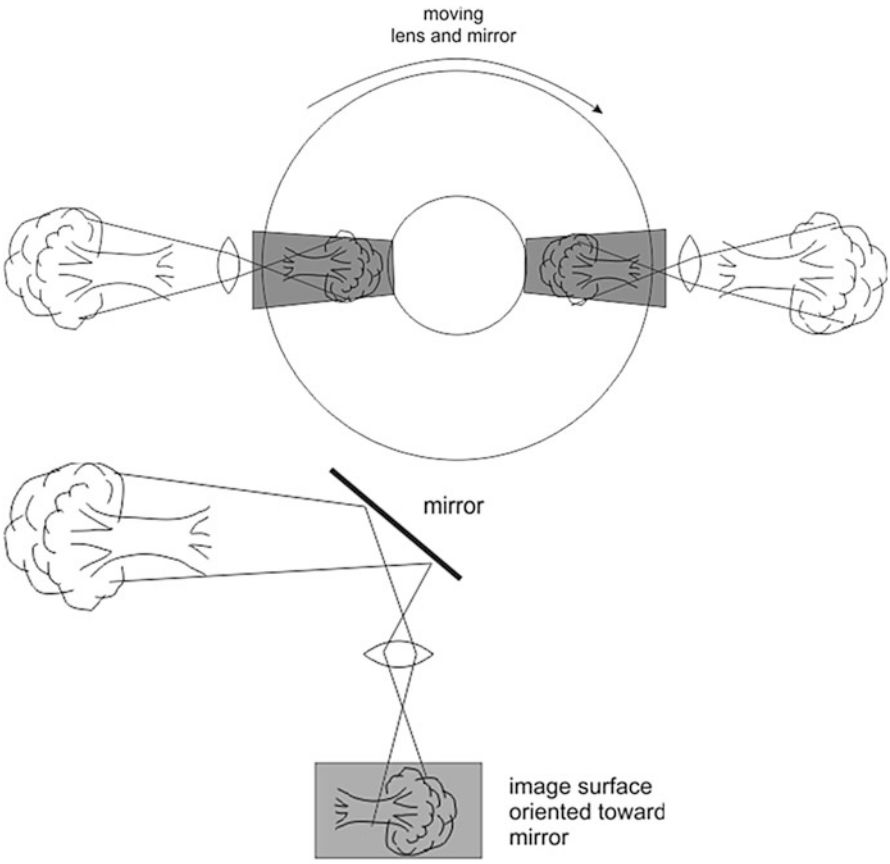


Fig. 2.38 An anamorphic panorama

Figure 2.38 shows an anamorphic panoramic system. A rotating lens and mirror paint the scene onto the image surface through a slit. The bottom of the scene is stretched relative to the top. The image can be displayed through a rotating lens/mirror system, thanks to the reversibility of ray optics. This style of imagery was practiced as painting during the Renaissance. The Uffizi in Florence has several examples of these anamorphic paintings that are viewed using a cylindrical mirror placed in the center of the image. These amazing paintings were also painted through the cylindrical mirror.

An alternative anamorphic panorama is created with a *fisheye* lens. A fisheye lens provides an extremely wide angle field of view. As shown in Fig. 2.39, the fisheye lens maps the scene both horizontally and vertically onto the horizontal image surface.

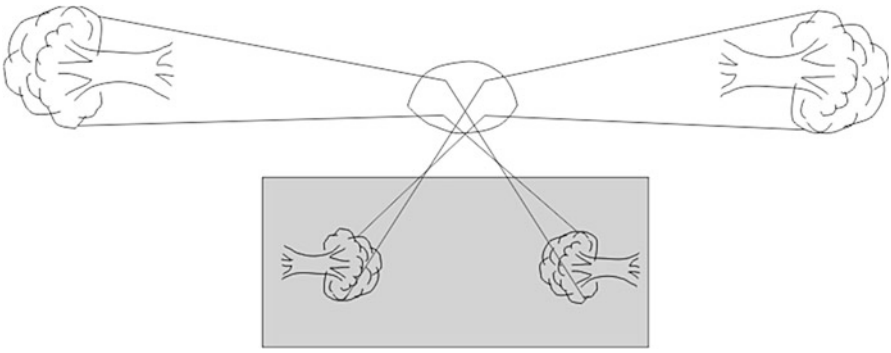


Fig. 2.39 A fisheye panorama

2.5.5 *Assessing Lenses*

We use several terms to describe related but distinct concepts:

- The ability to distinguish fine detail is known as *resolution*. We can quantify the concept of resolution.
- The term *acutance* is used in two different ways: specifically based on contrastor as a general, perceptual sense of sharpness. We will discuss acutance in more detail in Sect. 2.8.5.

An early quantitative definition for resolution is *Rayleigh's criterion*. Figure 2.40 shows a pair of slits, each projecting its own diffraction pattern. Rayleigh's criterion states that the minimum distance that can be resolved by these two slits is the distance between the peak of one diffraction pattern and the first minimum of the other.

The *Airy disk* is the luminance pattern created by an ideal lens with a finite aperture. It appears as a bright center with concentric, alternating regions of dark and light.

Rayleigh's criterion suggests testing resolution using *line pairs*—measuring the finest distance between two lines that the lens can resolve. The Air Force Resolution Test Chart can be used for resolution tests. The chart is carefully manufactured to provide precise line spacings and high contrast over a wide range of line pair spacings. The IEEE resolution chart was developed for analog television.

A more analytical method for image assessment is known as the *modulation transfer function* (*MTF*) [Nas08, Sch98]. Modulation refers to the variation between peak and trough in signals. Figure 2.41 shows a bar test pattern. We can plot the intensity of the bars as a function of position. The lens will determine how these bars are rendered in the image. A poor lens will spread the bars out, causing them to overlap. The result is a smaller difference between bright and dark regions in the image, which is represented by low modulation. A good lens will resolve the bars

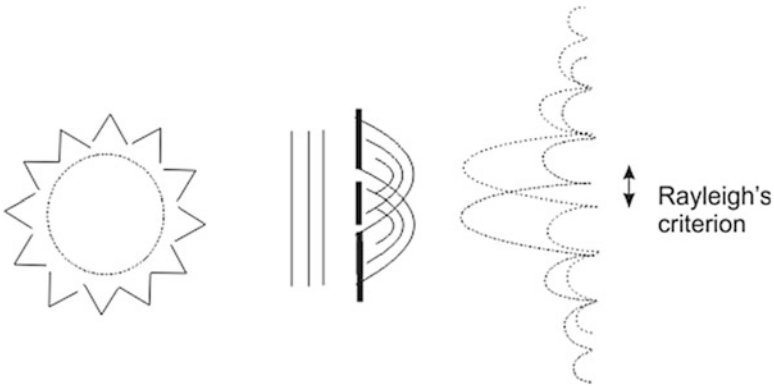


Fig. 2.40 Rayleigh's criterion

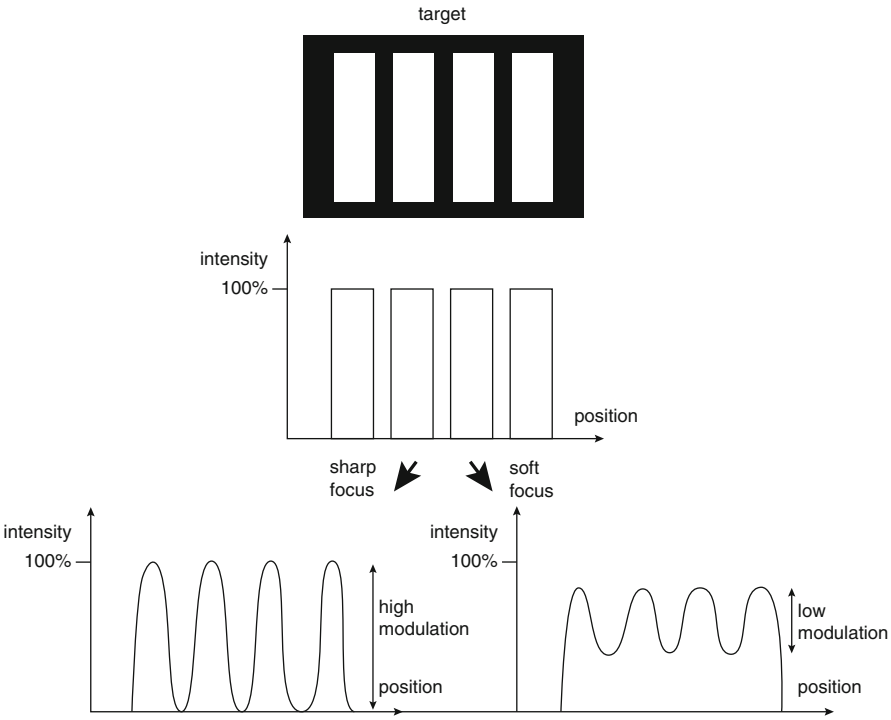


Fig. 2.41 Intensity modulation in images

sharply with less overlap between the light and dark areas, resulting in higher modulation.

We use *contrast* to measure modulation:

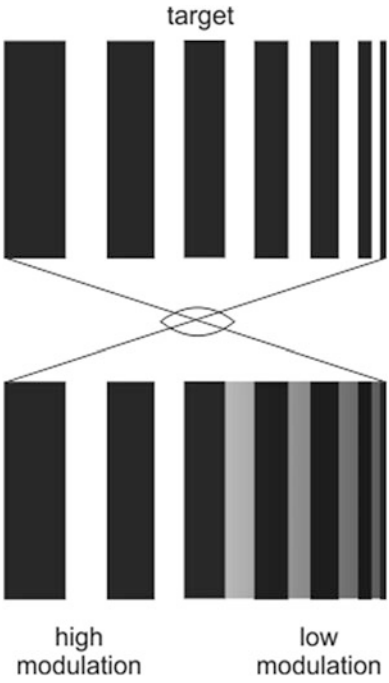


Fig. 2.42 Measuring modulation vs. spatial frequency

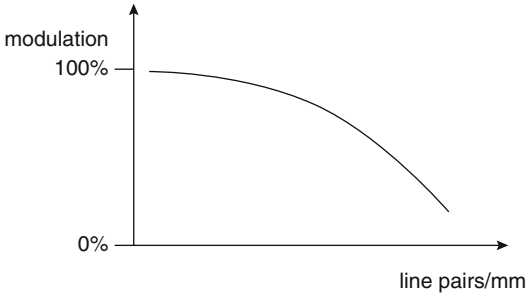


Fig. 2.43 An MTF plot

$$\text{Contrast} = \frac{\max - \min}{\max + \min} \tag{2.17}$$

We can use a bar chart with varying spacing to measure the modulation of a lens as a function of line spacing. In Fig. 2.42, the bar chart's spacing varies from wide on the left to narrow on the right. The lens produces an image with high modulation on the left, reproducing the bars well, and low modulation on the right, muddying the bars. If we plot contrast as a function of line spacing, we get the MTF plot of Fig. 2.43. The x axis of the plot is line pair spacing distance; the y axis is modulation level.

Subjective factors will always play a role in the selection of lenses. For the film *Wall-E*, the creative team carefully recreated the lens characteristics of Panavision lenses used for film as well as the look of film stock itself; they liked the look these aberrations and defects gave to the characters and settings.

2.6 Geometry and the Camera Model

For many purposes, we do not need ray tracing optics to understand what the camera sees. (We will consider an exception in Sect. 4.7 when we study lens correction algorithms.) A simpler model, the *camera model*, is geometric. It is primarily concerned with the relationships between the scene and the camera and makes use of only a very abstract model of the camera optics. We will start with an introduction to the geometric algebra we will use to build these models. We will then move onto the camera model itself.

2.6.1 Projective Geometry

Images are two-dimensional with points ranging over $[u \;\; v]$. The world is a three-dimensional *Euclidean space* with points $[x \;\; y \;\; z]$. We can use algebra to understand the relationships between these spaces: we need to be able to move within Euclidean space; we also need to map Euclidean space into the image space.

We use *homogeneous coordinates* [Car78] to simplify our manipulations. The term *homogeneous* refers to the fact that these coordinates do not assume a particular origin. The homogeneous coordinate system for 3D uses four dimensions to allow perspective transformations—lines in the 4D space map onto points in the 3D space. We can represent a 3D point in the 4D space as $[x\, y\, z\, 1]$; a point in the 4D space $[x\, y\, z\, h], h \neq 0$ represents the 3D point $[x/wh \;\; y/h \;\; z/h]$. A point in 3D space corresponds to the points along a line in 4D space. We can similarly construct a homogeneous coordinate system for the 2D image space $[u\, v\, w]$; it is easier and more consistent to map the homogeneous coordinates of Euclidean space into a homogeneous coordinate system for the 2D space.

A transformation matrix for the 3D homogeneous space is 4×4. For example, we can specify a translation in x as

$$\begin{bmatrix} ax \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \tag{2.18}$$

The upper-left 3×3 matrix is used for linear transformations. More generally, in the matrix

$$\begin{bmatrix} a_{11}\ a_{12}\ a_{13}\ p_1 \\ a_{21}\ a_{22}\ a_{33}\ p_2 \\ a_{31}\ a_{32}\ a_{33}\ p_3 \\ t_1\ t_2\ t_3\ 1 \end{bmatrix} \tag{2.19}$$

the a s represent linear transformations, the p s perspective transformations, and the t s translations. We can specify the perspective transformation for a distance d from the image plane as [Fol96]

$$\begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,1/d\,0 \end{bmatrix} \tag{2.20}$$

We can specify complex transformations as the product of several transformation matrices. The order is, of course, important. When analyzing the relationships between subjects and the camera, we often transform the subject's position to camera coordinates using a combination of rotation and translation.

For a given focal length f, we can describe the transformation of the 3D point into a 2D image point as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = {}^{f}\Big/_{z} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.21}$$

We can classify several types of transformations on images [Har03] as illustrated in Fig. 2.44. *Isometric* transformations perform rigid transformations:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta - \sin\theta\ t_x \\ \sin\theta\ \cos\theta\ t_y \\ 0\,0\,1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.22}$$

If the $1,1$ and $2,1$ entries are negated, the transformation reverses the image orientation. A *similarity* transformation combines isometric transformation and scaling:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{s}\ \cos\theta - s\sin\theta\ t_x \\ s\sin\theta\ s\cos\theta\ t_y \\ 0\,0\,1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.23}$$

An *affine* transformation preserves parallel lines, lengths of parallel line segments, and ratios of areas:

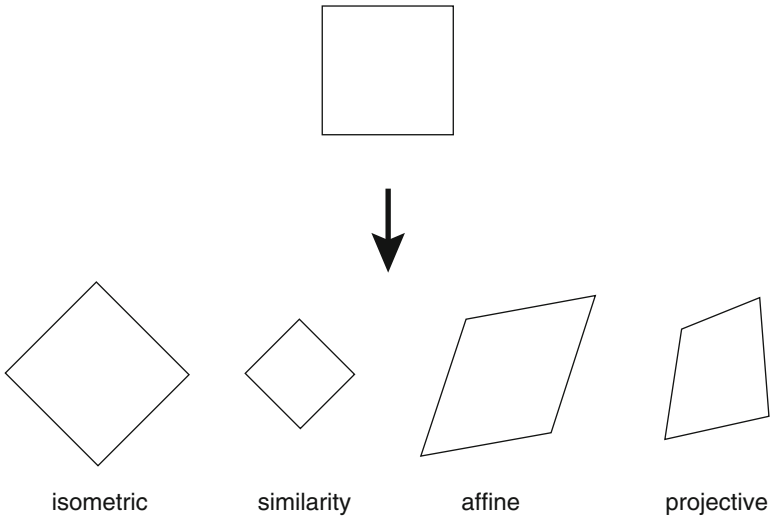


Fig. 2.44 2D geometric transformations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}a_{12}\ t_x \\ a_{21}\ a_{22}\ t_y \\ 0\ 0\ 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.24}$$

The most general form of 2D transformation is the *projective transformation*:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11}a_{12}\ t_x \\ a_{21}\ a_{22}\ t_y \\ v_1\ v_2\ v \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.25}$$

The upper block $a_{11}, \cdots, a_{22}$ performs rotation and scaling, the t_x, t_y terms perform translation, and the v_1, v_2 terms perform perspective operations.

A particularly important transformation for camera algorithms is a form of the projective transformation known as *2D homography* or as the *fundamental matrix*—three points in the source image lie on the same line if and only if they are also collinear in the transformed image. The homography has the form

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.26}$$

This homography has eight degrees of freedom. Given that the homography matrix has nine parameters, we need to impose a constraint on the parameters to ensure that the system has only eight degrees of freedom. A common assumption is that the homography parameters are related by a multiplicative constant.

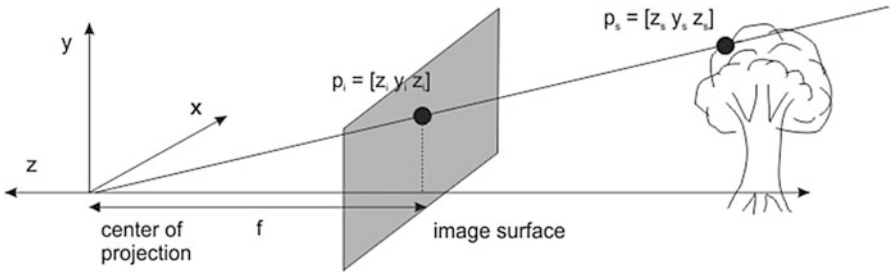


Fig. 2.45 The center of projection of the camera

2.6.2 The Camera Model

The *camera model* [Car78] describes how a pinhole camera projects the 3D scene onto the 2D image surface. It takes the form of a 4×3 matrix that transforms homogeneous points in three-dimensional space to homogeneous points on the 2D image surface. We can build up the camera model in several steps.

Figure 2.45 shows the basic model for the effect of the lens. Points on the subject p_s map onto points on the image surface p_i. The rays that connect the subject and image point pairs—a pair for each visible point on the subject—converge at the *center of projection*. The center of projection is located a distance f away from the image surface, f being the focal length of the lens. The z axis for the coordinate space positioned at the center of projection goes through the middle of the image surface. Longer focal length lenses put the center of projection farther away, resulting in less scaling of the subject to the image surface. The center of projection is not a physical entity, only an abstraction that allows us to build a very simplified model of the lens; the center of projection is not the lens' nodal point. The center of projection is behind the image surface, not between the image surface and the subject as the lens would be. However, this simple ray model allows us to capture the basic projection made by the lens without resorting to complex physical models of lenses.

The *projection ray* sets up two similar right triangles, one for p_s and the other for p_i. The image triangle is scaled by f relative to the subject triangle; we can write the positions of the image coordinates as

$$x_i = f\frac{x_s}{z_s}, y_i = f\frac{y_s}{z_s} \tag{2.27}$$

We can write these relationships in homogeneous coordinates for both the Euclidean and image spaces. This form is known as the *viewing matrix*:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = V \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} f\,0\,0 \\ 0\,f\,0 \\ 0\,0\,1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix}, x_i = u/w, y_i = v/w \tag{2.28}$$

All these positions are measured in world coordinate units.

We can generalize this simple model to include a simple set of *intrinsic camera parameters*: we can scale the image from world units to pixel units with a scale factor k; we can translate the image surface center from the axis with an offset $[x_0\, y_0]$; we can add a parameter s to skew the image surface frame in a simple affine transformation. These parameters are intended to model inaccuracies in the construction and operation of the camera: translation accounts for offsets between the image sensor center and the lens optical axis; skew helps to model effects of older video cameras; the scale factor helps to estimate the pixel spacing in the image sensor. The result is

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} kf\ s\ x_0\ 0 \\ 0\ kf\ y_0\ 0 \\ 0\,0\,1\,0 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = [K\ 0_{3\times 1}] \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \tag{2.29}$$

The left-hand 3×3 submatrix is known as the *calibration matrix* K; it has five degrees of freedom.

This model sets the image plane to be perpendicular to the optical axis. We can, however, use a camera model to describe the camera movements of Sect. 2. practical.composition. The center of projection refers to the position of the lens' optical axis; we do not have to worry about front board movements relative to the camera frame. This camera model assumed in the camera model does not take into account lens characteristics or depth-of-field. We can model rear board movements using a homography that describes the translation from the default image surface position (centered on and perpendicular to the optical axis) to its adjusted position. By measuring these movements relative to the lens board, we can take into account, for example, shifts of the front board (Fig. 2.46).

The position of the camera relative to a scene origin is a set of *extrinsic camera parameters*. In general, we need to translate and rotate from the camera center of projection to the world origin, as shown in Fig. 2—camera-to-world. We can write the translation from a point in the subject to image coordinates as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = K[R\ T] \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = M \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \tag{2.30}$$

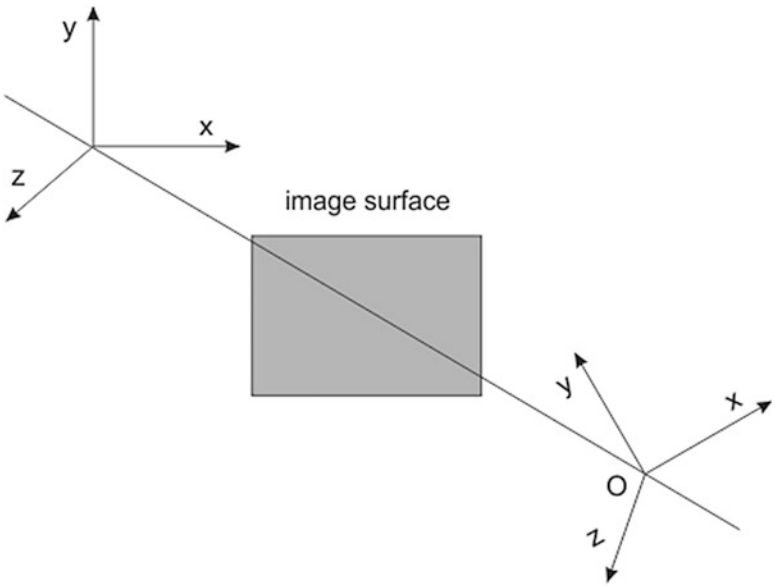


Fig. 2.46 Camera and world coordinates

where K is the calibration matrix, R is a 3×3 rotation matrix, and T is a 3×1 translation matrix. Their 3×4 product M is known as the *projection matrix*. It has 11 degrees of freedom.

2.6.3 Camera Calibration

Camera calibration experimentally determines the intrinsic and extrinsic camera parameters. Using a tape measure to find the location of the camera relative to the subject is slow and unwieldy; given the noticeable variations in camera manufacturing, published specifications for internal parameters should be treated with skepticism. Calibration algorithms extract the camera parameters from imagery.

Tsai's calibration methods [Tsa87] have been widely influential. His method made use of a *calibration target* shown in Fig. 2.47. The target consisted of a series of black squares placed on a flat target. Tsai developed two algorithms, a *coplanar* method which used one picture of a target and a *noncoplanar* method which used several pictures of the target at several different vertical positions. A variation on this approach is to put two of these targets at right angles. The corners of the squares are used as the *features*. The target is designed to create a large number of easy-to-identify targets, allowing for a good fit of the camera model in the inevitable presence of noise. Their position can be determined using standard algorithms such as Canny edge detection.

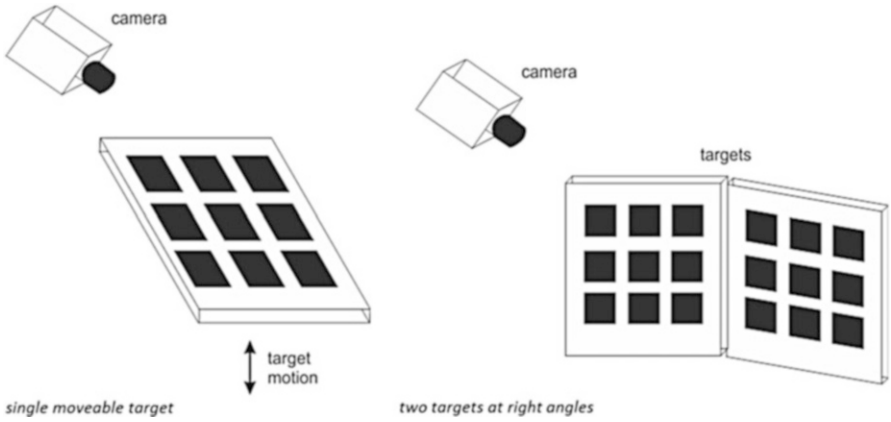


Fig. 2.47 Camera calibration targets

Tsai classified the parameters as group I, which required nonlinear estimation, and group II, which required only projective geometry equations. The coplanar and noncoplanar procedures first computed the group II parameters and then the group 1. Both methods first estimate the rotation and translation using linear methods, then estimate the remaining parameters using nonlinear methods.

The first step is to estimate rotation R and translation T. For each image feature point $\langle X_{di}, Y_{di} \rangle$, this linear equation can be formulated to describe its relationship to the feature in world coordinates at $\langle x_{wi}, y_{wi}, z_{wi} \rangle$:

$$[Y_{di}x_{wi} \;\; Y_{di}y_{wi} \; Y_{di} - X_{di}x_{wi} \; X_{di}y_{wi}] \begin{bmatrix} T_y^{-1}r_1 \\ T_y^{-1}r_2 \\ T_y^{-1}T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \end{bmatrix} = X_{di} \tag{2.31}$$

After this system of equations is solved, the rotation matrix parameters r_i and the translation parameters T_x, T_y can be found. A separate test needs to determine the sign of T_y. The rotation matrix and the focal length f are then computed from the r_is.

The focal length f and the translation T_z are estimated ignoring lens distortion using an overdetermined set of linear relationships between the feature points and the rotation and translation parameters. Given these initial estimates, the final values for these parameters as well as the distortion parameters κ_1, κ_2 can be found using standard nonlinear solution techniques. The target plane must not be exactly parallel to the image surface; if it is, the equations in this step will become linearly dependent.

The coplanar algorithm can be used when the scale parameter s is known. If it is not known, the noncoplanar method must be used. This method is broadly similar but must handle many more measurements and overdeterminism.

In the case of video, an additional intrinsic parameter is the frame rate. Relying on the manufacturer's specification is once again error-prone. Our group has seen frame-rate variations of 20% in consumer cameras; others have reported to us errors of 0.5% in their professional equipment, which amounts to a full frame every 6.7 *s*. Less attention is paid in the literature to frame-rate calibration, but we can measure frame rate by capturing a video sequence of a visible time reference; the length of the video will limit the accuracy of the calibration. Temporal calibration is critical when comparing the video from multiple cameras. We will discuss multicamera calibration in Sect. 4.multicalibration.

2.7 Image Display

Our perception of an image depends in part on how it is displayed. In this section, we will look at how several common types of displays work and how to manage the display process.

Images can be displayed either on paper or on electronic devices. The *ink jet printer* [Nie85, All85, Bha85] provided printed computer output that was both high quality and low cost. The inkjet printhead boils a tiny amount of ink, causing the ink to spit out to the paper. The ink droplets are a picoliter in size, and their position on the page can be very accurately controlled. A scanning head lays down a column of ink jets at each horizontal position on the page; multiple scans build up the complete image. Modern archival-quality inkjet printers may use a dozen different ink colors to produce a wide range of colors.

Three major electronic display technologies are in wide use today: LCD, OLED, and DLP. Table 2.1 compares the characteristics of these types of displays.

The basic element of an *LCD* (*liquid crystal display*) is a *light valve*. Liquid crystals can be oriented in the presence of an electric field. Their orientation can be used to change the valve's polarization properties and the amount of light through the valve. Color displays are built by adding color filters to adjacent pixels. The LCD element does not produce its own light but instead depends on reflected or transmitted light. The light valve cannot be made completely transparent, limiting its dynamic range. It may also take time to change the value of a pixel, resulting in some image persistence. Some displays use *quantum dots* to generate light for LCDs. A quantum dot structure's output wavelength depends on the size of the dot structure, which allows the wavelength of the generated light to be precisely controlled. A quantum dot can be pumped from a wideband light source to produce light with more precise wavelength characteristics.

OLEDs (*optical light emitting diodes*, also known as *amorphous OLEDs* or *AMOLEDs*), in contrast, generate their own light at each pixel. They make use of special organic materials that act like the silicon semiconductors used in chips. OLEDs are very bright and can produce vivid colors. However, they degrade with use quickly enough that each pixel in the display contains a circuit that measures the

Table 2.1 Displays: Characteristics of electronic displays compared

Display type	Dynamic range	Gamut	Persistence
LCD	Medium	Medium	Medium
OLED	High	Large	Medium
DLP	High	Large	Low

characteristics of the LED and adjusts the circuit to compensate for changes. OLEDs exhibit some image persistence.

The *DLP* (*digital light processor*) makes use of a controllable reflector at each pixel. An internal light source shines on each reflector, which can be oriented to either reflect out toward the lens or inward to a light baffle. The intensity of the pixel is controlled by blinking the mirror several times, with more blinks corresponding to a brighter pixel. Color images can be displayed in either of two ways: three DLP units, one for each primary color, or with a color wheel that alternates the color source between the primaries. Mirrors can be manufactured to be extremely efficient reflectors, so the brightness of the displayed image is primarily limited by the brightness of the internal light source.

The useful resolution of a display is ultimately limited by the acutance of the visual system. As with image capture, the ideal viewing distance for a displayed image is equal to its diagonal. We saw in Sect. 2.eye that the resolution of the eye is about 1 arc minute. This translates to a maximum useful resolution of about 4K lines at the standard viewing distance.

Virtual reality requires extremely high spatial resolution and frame rates. HD frame resolutions are minimal, with some systems using 4K displays. Frame rates run at 90 frames/sec or higher.

3D is primarily used today for motion pictures, although 3D still images were popular in both the nineteenth and twentieth centuries (e.g., the *ViewMaster*). 3D image display depends on *disparity* between the position of objects at the two eyes as produced by a pair of images, each shot a small horizontal distance from the other. Still image viewing systems could use a separate optical path for each eye. Theater or home-theater 3D systems must use a single display for both images. The 1950s 3D movie craze (*Creature of the Black Lagoon,* etc.) was based on monochrome movies; in this case, the left and right images could be projected through red and cyan filters, with the images separated at the eyes using red/cyan glasses. Modern color 3D systems polarize the two images and use glasses for which each side has a different polarization.

Early computer display software did not distinguish between the colors and luminances in the image to be displayed and their representation on the screen. As a result, the same image could look very different when displayed on different devices. A device's *gamut* is the range of colors it can reproduce; it is defined relative to a color space such as the CIE color space. As shown in Fig. 2.48, several different gamuts have been defined for different types of devices: sRGB for CRTs, Adobe RGB for printers, and Red.2020 for UHDTV.

Color management systems are designed to separate the characteristics of a particular computer system from the image being displayed and provide *device-*

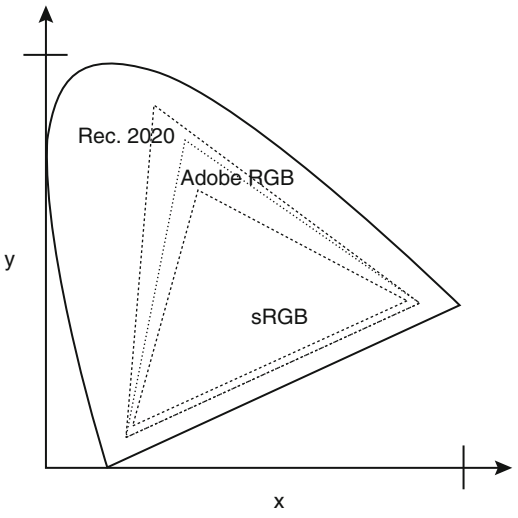


Fig. 2.48 Several gamuts compared to the CIE color space

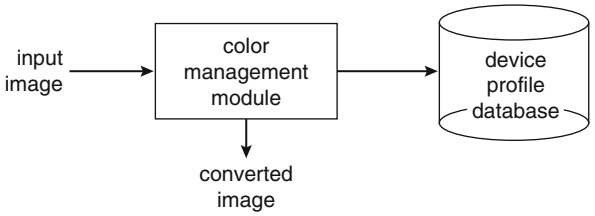


Fig. 2.49 A color management system

independent color. As shown in Fig. 2.49, a color management system has two major components: a set of *device profiles* that describe the gamuts of devices being used and a *color management module* that translates between devices. Given an input image in one gamut, the color management module will produce a new version of the image converted to the gamut of the output device. Many image formats encode gamut information—for example, whether the image was taken in the sRGB or Adobe color space.

Moviemakers take extreme care with the management of colors. A *color lookup table* (*LUT*) is used to give the color used to represent each possible pixel value. Filmmakers will fine-tune the entries in their LUTs to adjust the rendering of their images onto the screen. This approach can be seen as a specialized form of color management.

2.8 Practical Image Capture

In this section, we look at the real-world capture of photographic images: exposure, image composition, lighting, and perspective. We close with sections on image quality assessment and the design of shutters and irises.

2.8.1 Exposure Settings

For both still and moving images, we need to be able to capture a still image. We do not capture images instantaneously—we need to wait a certain amount of time to gather enough light. But we need to control the *exposure* of our image surface to the image to capture an image. In the case of still images, we do this once. In the case of video, we capture a sequence of still images in rapid succession.

Image sensors vary widely in their sensitivity to light. In order to create a usable image, we need to measure the sensitivity of our image sensor so that we can determine the required exposure. The standard measure for image sensor sensitivity is known simply as *ISO*, although that is an abuse of terminology (the acronym refers to the International Standards Organization, which has issued a particular standard on the subject). ISO numbers are known as *speeds* with higher numbers indicating more sensitive image sensors. Given the logarithmic relationship of luminance to perceived brightness, we are interested in doublings and halvings of ISO speeds: 100, 200, 400, etc.

We control the actual exposure using a combination of two mechanisms: the *iris* (also known as the *aperture*) and the *shutter*. As shown in Fig. 2.50, the iris is part of the lens and forms an aperture that controls the amount of light through the lens. The shutter controls the duration with which the image is exposed to the image sensor; the shutter may be either in the lens or in front of the image sensor, although in most digital cameras it is at the image sensor.

Shutter settings, referred to as *shutter speeds*, are in fractions of a second. As with ISO ratings, we are interested in powers of two. Typical shutter setting values are 1/30, 1/60, 1/125, 1/250, etc.

Aperture settings are also designed to change the light through the lens in multiples of 2. The illumination passing through the iris depends upon the area of the iris, which makes the numbering system more complicated. We measure the iris size relative to the lens' focal length—this allows us to use consistent iris settings even if we change to a lens with a different focal length. The iris setting is known as an *f-stop* and is pronounced simply as, for example, "*f 8*." Typical f-numbers are *f*/2.8, *f*/4, *f*/5.6, *f*/8, etc. (These numbers are iris diameters generated from $2^{i/2}$.) A larger number in the denominator refers to a larger aperture setting. Professional cinematographers still use *T-stop* terminology, but this term is not generally used elsewhere; a T-stop includes losses in the lens as well as the aperture. Zoom lenses may change the amount of light they throw onto the image surface while zooming; t-stop settings take those changes into account.

As we change focus, the distance from the lens to the image surface changes, changing the irradiance per unit area on the image plane. We must adjust exposure. . ..

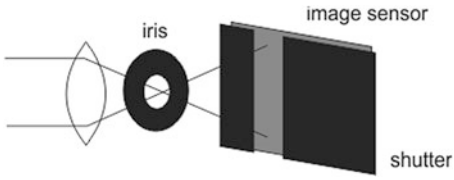


Fig. 2.50 Iris and shutter

Remember that the lens' focal length determines the angle of view of the subject that it paints on the image surface. We identify lenses with two numbers: its focal length and its *maximum* f-stop. The maximum f-stop is the widest aperture setting available on the lens. A lens with a wider maximum f-stop is referred to as a *fast lens*. Although fast is good, a large maximum aperture often brings other trade-offs for lens design that may result in some less desirable characteristics for the lens.

Given an exposure, we can determine the shutter speed and aperture required. The exposure measurement system typically reports directly the shutter speed and aperture. The luminance value used to determine the exposure often is not reported. However, we have a degree of freedom in setting the exposure due to *reciprocity*. The total amount of light falling on the image surface is what matters, so by doubling the f-stop and halving the shutter speed (or *vice versa*), we maintain the same exposure. For example, an exposure of *f*/5.6 at 1/125 sec is equivalent to *f*/8 at 1/60 sec and to *f*/4 at 1/250 sec. Reciprocity gives a great deal of freedom: faster exposure times allow us to reduce motion blur; smaller apertures give us greater depth-of-field. We can therefore select the combination of shutter speed and aperture best suited to the type of image we want to capture. Reciprocity also allows us to refer to changes in exposure in terms of *stops*—for example, increasing or decreasing the exposure by one stop. We also mix our metaphors with phrases such as "one stop slower" or "one stop faster." (Reciprocity may fail in extreme cases in some types of image sensors, particularly very long exposure times.)

We sometimes refer to exposure in terms of *exposure value* or *EV*. An EV reading describes luminosity, so we can turn it into any combination of shutter speed and aperture we want. Using EV allows us to succinctly describe light intensity without adding caveats about reciprocity.

2.8.2 Which Exposure Setting?

However, determining what exposure is best to capture a given scene is not always simple. We face two problems: we need to choose how to represent luminances in the subject as tones in the final image; and we need to know the subject to know what luminances in the scene to measure.

The first reality we need to face is that we can make any part of subject appear to be white or black in the final image simply by changing our exposure. Figure 2.51 shows three different exposures of a standard reference card known as a *gray scale*. The middle exposure is a nominal value; the top one EV more exposure; and the

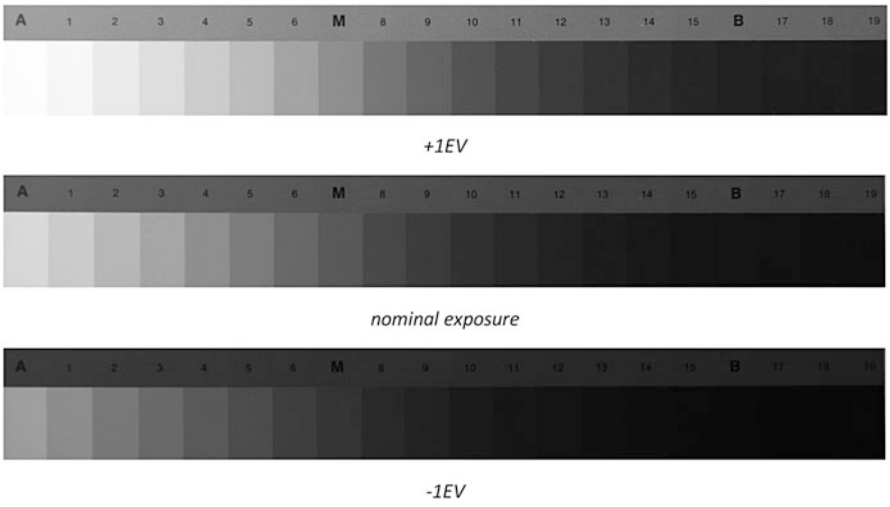


Fig. 2.51 Changing exposures changes the tonal representation of the subject

bottom strip received one stop less exposure. Changes in exposure result in the steps in the gray scale changing their tones in the final image. This example shows us the limits of representation of the world by photos. When we look at an image, we cannot say that a given object in the scene was white or black. We can only say that it was rendered so in the image.

We therefore face a choice in how we render the subject as tones in the image. Although image manipulations that we will discuss in Chap. 4 will allow us to change the tones of parts of the image selectively, exposure will change all the tones in the subject in lockstep as we increase or decrease the exposure.

We will use a *light meter* to measure light intensity so that we can determine our exposure. Ultimately, we will use the camera itself as a light meter, but imagine for a moment that our meter is a separate instrument. A light meter is calibrated to give an exposure that results in a mid-level gray image. Unfortunately, manufacturers do not agree on the definition of mid-level gray. Different cameras or light meters may be calibrated to different standards, resulting in slightly different results. We have no clear, well-accepted definition of a mid-level gray. We will use the term *mid-level gray* without assuming a particular meaning. Since most scenes do not provide an obvious mid-level gray, we can use a *gray card* as a reference. Many gray cards are printed to an 18% reflectance. Some authors, including Adams, have equated the 18% reflectance value to mid-level gray, but the lack of a common standard for calibration means that different light meters may render that reflectance to different gray levels.

Figure 2.52 shows two different techniques we can use to find a reference exposure that will render the image with standard tonality. *Incident metering* measures the light falling on the subject. *Reflective metering* measures the light reflected from the subject toward the camera.

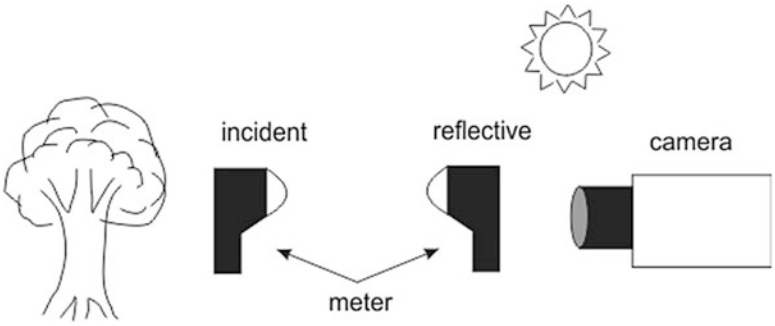


Fig. 2.52 Incident and reflective metering

Fig. 2.53 Mis-exposure due to the reflective characteristics of the subject

If the light reflected from a subject is measured to directly find an exposure, the result may be a misrendering of the subject. A simple example is a piece of white paper, shown in Fig. 2.53. The image was exposed using a reflective metering off the paper. The result was an exposure that rendered the paper mid-level gray.

In order to determine the proper exposure for an image, we need to know the subject and have some idea of how we want to render it. For example, many photos of people are shot against bright lights or windows. An exposure based on the total amount of light in the scene results in the exposure favoring the background but leaving the face very underexposed and hard to read. Figure 2.54 shows two selfies: one exposed primarily for the face tones and another at -2EV relative to the face-weighted exposure. The face-weighted exposure renders the face well, but the scene outside the window is blown out. Reducing the exposure gives more detail of the background, but the face is now darker.

The Zone System of Ansel Adams [Ada02B] helps us understand the process of choosing an exposure. The Zone System was formulated in the film era, but it still offers us many lessons. Adams divided tones into 11 zones labeled with Roman numerals: 0 for black through X for white. Zones are separated by one stop. The zones correspond to the range of luminances that can be captured by film; some image sensors provide somewhat wider dynamic range, as we will discuss in Chap. 3. Figure 2.55 shows some regions of a photo labeled with their zones.

Previsualization is Adams' term for thinking about how you want the photo to look and deciding on the tones to be used in the image. If you choose a part of the

Fig. 2.54 Photographs of a face with backlighting

image and decide what tone should be used for it, you can then choose your exposure accordingly. Zone V corresponds to 50% gray, so a meter reading of that part of the subject will produce a 50% gray tone in the image. You can change the zone at which the object is rendered by changing the exposure: Zone VII, for example, requires increasing the exposure by two stops and Zone III is reached by reducing the exposure two stops.

If exposure was our only tool, picking the zone for an object in the image would determine the tones of everything else in the image as well. We will see in Sect. 4.3 that we can use image manipulation tools to compress or expand the range of tones; Adams accomplished the same goal by changing the development time of film. Figure 2.56 shows an image before and after its tonal range has been adjusted.

Adams recommends that photos be previsualized to contain both a solid white and a solid black so that the eye has proper reference points. His recommendation is consistent with the anchoring heuristic that we introduced in Sect. 2.3, which suggests that the visual system adjusts itself relative to the darkest and lightest parts of the image.

We would like our camera to help us find good exposures for our image. In the case of simple images, we would like the camera to make all the decisions—we want the camera to previsualize the image for us and decide what camera settings achieve the photo we desire. We would like its assistance when we take on more of

Fig. 2.55 Example zones in a photo

the artistic responsibilities to ourselves. We will explore these topics in more detail in Chaps. 3 and 4.

2.8.3 Color Temperature

Just as we can render luminances in the image into many different tones, we can also render color in different ways. The color temperature of the illumination influences the color received by the image sensor—the subject cannot reflect color wavelengths that it does not receive from the illumination.

Much as we can affect tonalities by changing exposure, we can change the captured color of an image by changing its illumination. Common forms of lighting—fluorescent, incandescent, and tungsten—operate at different color temperatures. A light source with the characteristics of daylight can be created from LEDs, either by using red, blue, and green LEDs or a single LED and phosphors to generate the lights of the other colors. Control circuits can be used to vary the relative outputs of red/green/blue LED arrays to provide a variable color temperature light source.

before

after

Fig. 2.56 An image before and after tonal range adjustment

In order to render the image naturally, we need to correct for the color temperature of the illumination. Many cameras allow you to select the type of light to provide a preset color correction. We will look at automated color temperature correction in Sect. 3.post.whitebalance.

2.8.4 Image Composition

The *composition* of an image—the arrangement of elements within the image—is an important aspect of its previsualization. Poor exposure choices may leave parts of the image with undesirable tones. We want to choose exposure to make sure that important subjects in the image are rendered in a suitable tone.

Unfortunately, no simple rule tells us where the subject of a photograph should be. As we saw in Sect. 2.3, the visual system scans the scene to build up its view. Our retina is incapable of viewing a large area at high resolution. As a result, good photographs (an admittedly subjective term) give the eye several interesting things to look at in different parts of the photograph. Those elements may be people, recognizable objects, or simple textures such as the leaves of a tree.

A simple and surprisingly effective rule for the placement of compositional elements is the *Rule of Thirds*, shown in Fig. 2.57. Divide the image into thirds both vertically and horizontally, then place elements at some or all of those intersections. The Rule of Thirds gives us several natural positions to spread interesting elements, neither too far apart nor too close to each other or the borders.

The *Golden Ratio* is a more sophisticated and elegant rule for composition that is widely used in all types of art. In Fig. 2.58, the ratio a/b is equal to the ratio $a + b/a$. This allows us to repeatedly subdivide the rectangle into smaller and smaller pieces that all satisfy the Golden Ratio.

Many aspects beyond simple placement of interesting objects play into composition. Depth-of-field is an important cue for attention. We can control depth-of-field using the aperture: wide apertures give less depth-of-field, while narrow apertures give more. Figure 2.59 shows photos of a still life taken at several different apertures. The shot taken with the wider aperture renders the flowers at the back of the bush somewhat out of focus and the porch as very unfocused. The small aperture shot renders the background with much finer detail. Deep depth-of-field is often important for documentary or technical photographs, but in some situations, it results in distractions from the main subject. The bokeh of out-of-focus objects can also be used as an artistic element. Some cameras that use stereo information can be used to rerender backgrounds with bokeh.

Perspective is a natural phenomenon that can also be used to dramatic effect. Figure 2.60 shows a simple example: the two lines are parallel, but as they move farther away from the camera, their separation becomes a smaller fraction of the angle of view. As a result, they appear to move together. At the horizon, they appear to join together at the *vanishing point*.

Perspective effects can be seen at any orientation. Figure 2.61 shows examples of horizontal and vertical perspective. Perspective effects are hard to perceive directly, thanks to the constancy mechanisms of the human visual system. Careful and relatively slow observation of a building, for example, can reveal its vanishing points, but we normally do not pay attention to small perspective changes.

We can control some aspects of perspective as well as sharpness by controlling the relative position of the image surface and lens. We have assumed that the lens

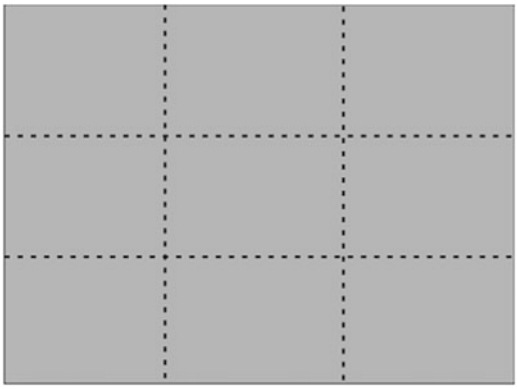


Fig. 2.57 The Rule of Thirds

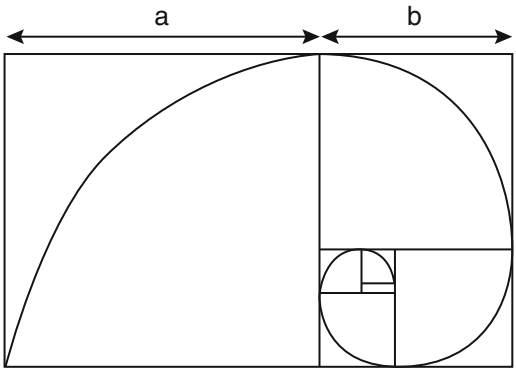


Fig. 2.58 The Golden Ratio

and image surface are centered on a common axis, but this does not need to be the case. A *view camera* provides several degrees of freedom for both the lens and image surface; these movements can be used to manage perspective. We will discuss the mathematics of perspective correction in Sect. 4.9.

We can also use camera motions to control sharpness. The *Scheimpflug rule* [Kod88] is illustrated in Fig. 2.62. The image surface, lens plane, and subject plane all meet at a single point. As a result, the image plane captures the maximum depth-of-field.

Lighting provides important depth cues as well as drama. Highlights and shadows can illustrate the shape of an object. Rim lighting—a light behind the subject—provides a glowing boundary for the subject.

Composition for video generally follows the rules for still images while adding additional techniques. The basic types of cinematic shots are named relative to the scene and the people in it: an *establishing shot* shows the entire scene; a *two-shot* shows two people; a *medium shot* shows the upper portion of a person; and a *close-up* concentrates on the person's face. Cinematic tradition holds that D. W. Griffith on *Birth of a Nation*. The films of John Frankenheimer, for example *The Manchurian Candidate*, provide examples of strong composition as a dramatic tool. Motion

1/200 sec, f/7.1

1/40 sec, f/16

Fig. 2.59 Depth-of-field at several different apertures

picture scenes are often shot with shallow depth-of-field to encourage the viewer to avoid visual distractions.

Camera movement—horizontal movement is known as *panning*—can be used both to reframe the scene and to provide drama. Hitchcock's *Rope* was composed entirely of 10-min shots, each shot consuming an entire reel of film. Zoom lenses of sufficient quality for cinematic use became available in the 1960s, leading to a decade of shots in which zoom was used to dramatically change framing.

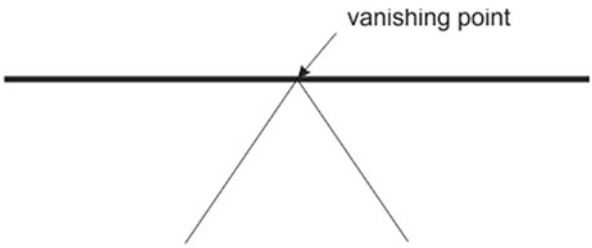


Fig. 2.60 The vanishing point of a pair of parallel lines

Fig. 2.61 Examples of horizontal and vertical perspective

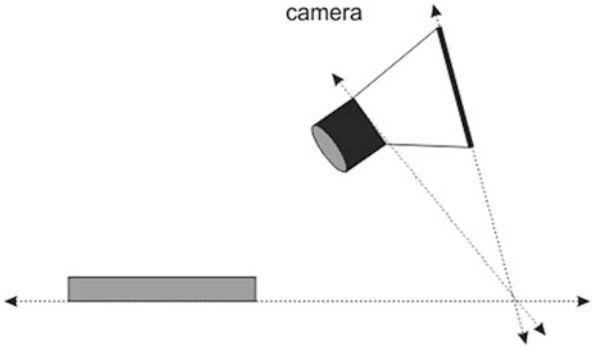


Fig. 2.62 The Scheimpflug rule for depth-of-field

Careful planning of shots helps to give the viewer a consistent point of view. The *180 degree rule* is a simple example. As shown in Fig. 2.63, placing the camera on one side of the subjects gives them one relative position, in this case actor A to the

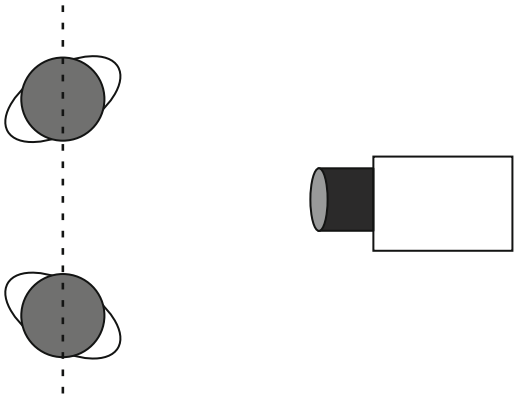


Fig. 2.63 The 180 degree rule

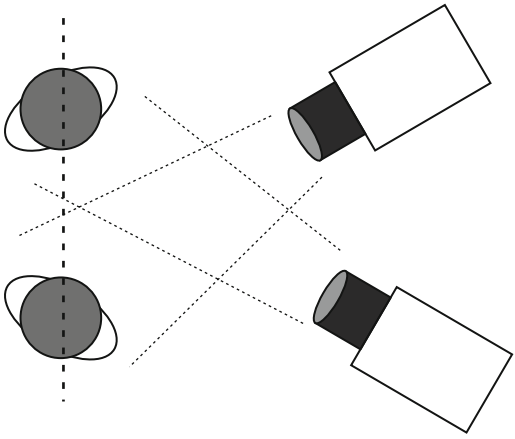


Fig. 2.64 Shooting dialog from complementary positions

left of actor B. If the camera were moved to the opposite side of the line, actor B would be to the left of actor A. Figure 2.64 shows the placement of cameras to capture a dialog between two actors. Actor A is shot by the lower camera, while actor B is shot by the upper camera. The positioning of the camera mimics to some extent the position of the other actor to give the sense that each actor is talking to the other.

Transitions are critical elements that help to distinguish cinema from still photography. Films are composed of many individual shots that are composed together. Several types of transitions can be used, each with their own application and meaning:

- Cuts move immediately from one shot to the next.
- Fades may go out to black (or some other color) or come into the scene from black.
- Dissolves overlap frames from two shots for a short interval, with one fading out while the other fades in.

- Wipes move a line or other shape across the screen that provides a boundary between one shot's frames and the next.

Montage was one of the fundamental discoveries of cinema. Montage was discovered by Soviet filmmakers in the 1920s who found that viewers inferred meaning from the juxtaposition of two shots. The *Kuleshov effect* demonstrates the phenomenon. The experiment combined a shot of an expressionless actor alternating with three other shots: a plate of soup, a girl in a coffin, and a woman on a divan. Audience members who saw the sequence believed that the actor's expression was different in each case even though all three shots of the actor were the same.

2.8.5 Image Quality Assessment

The assessment of image quality is no simple matter. Not only does the quality of an image depend on every component in the imaging chain—lens, image sensor, and display—but it also depends on the characteristics of the human visual system.

The modulation transfer function of the eye changes with illumination level [Sch64]. Lower illumination levels result in an ability to resolve somewhat lower spatial frequencies.

Katz's formula [Cox66] suggests that resolutions of components of a system be combined as

$$\frac{1}{R_{tot}^2} = \frac{1}{R_a^2} + \frac{1}{R_b^2} \tag{2.32}$$

This formula is heuristic and not based on detailed vision science.

Higgins and Jones [Hig52] found that resolving power does not correlate well with the subjective experience of sharpness.

The term *acutance* is often used as a measure of picture contrast. The most general definition of acutance at a given point is the gradient of image density. We can also calculate acutance using samples. Consider the response to a grating—a spatial square wave—as shown in Fig. 2.65. We choose a region of the response curve of length l from a local maximum to the next local minimum. We divide the region into equal-sized strips of width $a = l/n$. Each strip has a density D_i and a density slope (change in density from left edge to right edge) r_i. Acutance is defined as

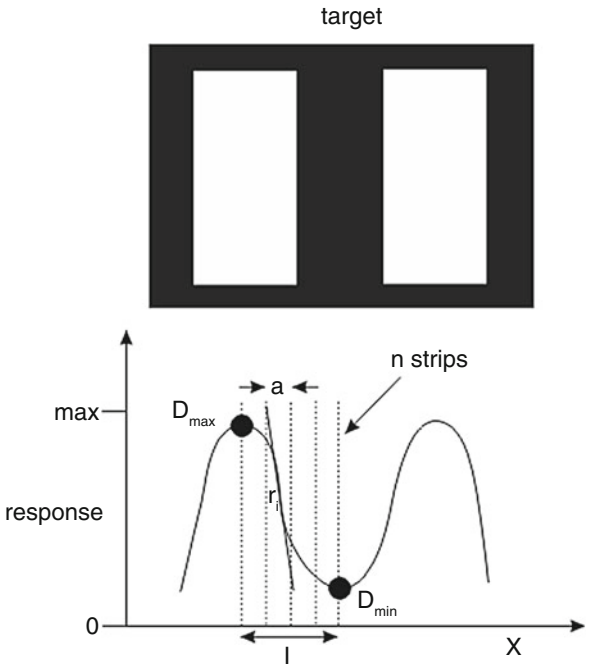


Fig. 2.65 Measuring acutance

$$A = \frac{1}{n}|D_{max} - D_{min}| \sum_{1 \le i \le n} r_i^2 \tag{2.33}$$

where D_{max} and D_{min} are the maximum and minimum density values, respectively. Kodak defined a formula for the combination of acutance and resolving power R [Cox66]:

$$K = A\left(1 - e^{-cR}\right) \tag{2.34}$$

If resolving power is expressed in lines per mm, then $c = 0.007$. Both acutance and resolving power are important to an overall sense of sharpness. But given that the visual system prefers to have solid black and white areas in an image, acutance is an important feature of perceptual acceptability.

Subjective quality factor (*SQF*) [Gra72] is widely used to measure perceived quality of a rendered image. SQF compares image metrics to a subjective and undefined notion of quality. Granger and Cupery used a panel of viewers to evaluate quality of images; they did not define quality for the panelists in order to elicit a natural and comprehensive reaction to the test images. They defined SQF as

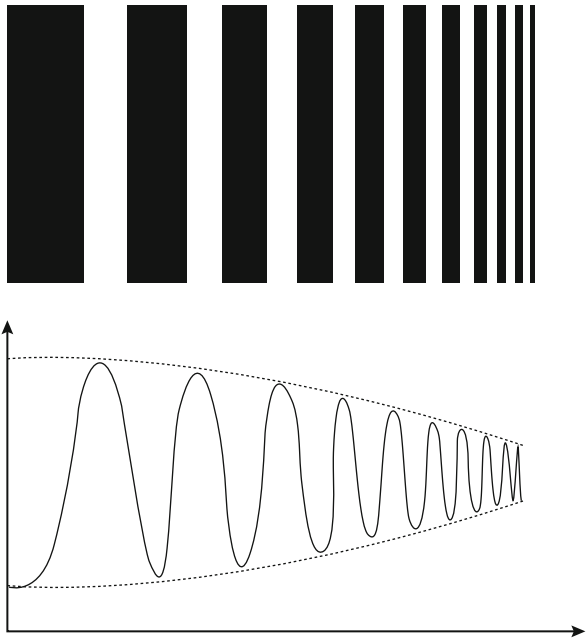


Fig. 2.66 Response to a logarithmic grating

$$SQF = K \int_{10}^{40} \int_{0}^{2\pi} |r(f,\theta)| d(\log f) d\theta \tag{2.35}$$

The outer integral is taken over the range $10 - 40$ *lines/mm*, the range in which the visual system has a high MTF as shown in Fig. 2-eye-mtf. The inner integral is performed in polar coordinates to take into account both horizontal and vertical resolution: f is the spatial frequency in *cycles/mm* at angle θ; $d(\log f)$ is the radial optical transfer function (MTF + optics); and K is a normalizing constant determined by integrating at $d = 1$. Granger and Cupery developed a log periodic test chart to evaluate SQF and quality. As illustrated in Fig. 2.66, for a lens of typical quality, the target's modulation is proportional to the system MTF.

The IEEE Standard for Camera Phone Image Quality (CPIQ) Working Group has developed its own definition of acutance. Both metrics combine the modulation transfer function of the imaging system, the contrast sensitivity function of the human visual system, image display height, and viewing distance.

2.9 Summary

Images are the result of manipulating light—we cannot see or take pictures without an optical system. Some simple characteristics, such as focal length and maximum aperture, give us quite a bit of information about a lens and the nature of the image it throws. We can use our understanding of lenses and imaging to throw a wide range of images of a scene, all with different characteristics. There is, however, single no best image to represent a scene. The image we want to capture depends on the uses to which the image will be put and the equipment we use to capture and display the image. Photographers can improve the quality of image they create by *previsualizing* the image they want and determining how to produce that result. We will spend the next two chapters studying how modern digital cameras can *autoprevisualize* images to create good-looking results with little or no input from the photographer.

Further Reading

Feynman [Fey10] is the go-to reference for physics. The *Focal Encyclopedia of Photography* [Per07] is an excellent reference on photography. Palmer [Pal99] provides a deep introduction to vision science. Arnheim [Arn74] relates vision and perception to art. *Imaging and Perception* provides a number of insights into the photographer's relationship to perception. The argument on the relationship between quantization and read noise is a very simplified version of a proof by Abbas El Gamal.

Chapter 3
Image Capture Systems and Algorithms

3.1 Introduction

This chapter considers the design of cameras and all the processes that are required to perform the initial processing of an image. We will concentrate in this chapter on algorithms that provide traditional photos, such as sharpening and compression. Imaging chain algorithms must be designed for efficiency. We measure efficiency along several axes:

- *Execution time*. Cameras—both still and video—are real-time systems. We care about the rate at which we can capture, process, and store images. Algorithms must be designed to run fast. We are also concerned about variations in their execution time, which can require additional buffer memory that imposes other costs and limitations.
- *Energy and power consumption*. Energy and power are related but distinct concerns. Energy is important because most cameras are battery-powered; lower energy per consumption per image results in more images per battery charge. Energy-efficient algorithms and systems must avoid unnecessary or duplicative work. Power consumption—energy per unit time—is important in large part because of thermal requirements. Power consumption results in heat. Thermal power dissipation is the primary limitation on performance in high-performance computer systems [Wol17]. Heat generated in a camera can also affect sensor performance—most device and circuit noise increases with temperature, typically exponentially.
- *Memory bandwidth and capacity*. Multimedia algorithms are memory-intensive. Memory and mass storage devices can absorb and produce data at limited rates. High memory access rates can limit system performance; it can also drive up energy and power consumption. We are also concerned with the total memory usage of an algorithm. Certain parts of the imaging pipeline, particularly those near the image sensor, provide only constrained amounts of memory. Sloppy use of buffer memory can, for example, limit the number of images in a burst.

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5_3

This chapter concentrates on the capture of images in digital cameras. We will start with a review of the camera design space. We will then analyze the design and characteristics of image sensors. We will next look at algorithms used preexposure and postexposure. We will then consider the computer architectures required for cameras. We will consider image characteristics and multicamera systems. We will close with a second look at trade-offs in camera design, based on our more nuanced understanding of the camera design space.

3.2 The Generic Camera Architecture

Figure 3.1 shows an architecture for a generic digital camera. While simplified—particularly in the case of memory—this block diagram shows the major components of a camera that applies to a broad range of realistic camera designs. It also applies to both video and digital still cameras.

The image capture subsystem renders an image onto the image sensor. The image capture unit may or may not include mechanical elements for focus and zoom, but most will include an iris as well as an electronic or mechanical shutter. A filtering engine performs early processing steps on the image. The results are then fed into a compression engine. The results are written to a mass storage device; they may also be rendered onto the display. More advanced cameras also include an image processing unit which may include a digital signal processor (DSP), a graphics processing unit (GPU), and specialized accelerators. A host processor or processor controls camera operation. A display is used to preview images, display captured images, and support the user interface. Modern digital cameras rely on sophisticated multiprocessors to perform their complex processing.

An important component of many digital cameras, particularly small ones, is a speaker through which the camera plays camera sounds. Users still expect cameras to make the sound of a mechanical shutter and film advance even though they do not have these mechanisms. Many cameras play prerecorded sounds to enhance the user's camera experience.

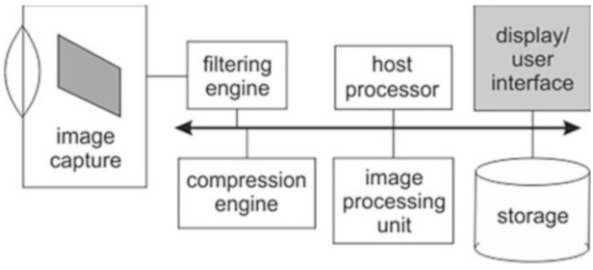


Fig. 3.1 A generic digital camera architecture

3.3 The Camera Design Space

The space of cameras that can be designed is large and multidimensional. To make good choices in the design of cameras, we need to understand the nature of the design space and the constraints that determine its shape.

3.3.1 Trade-Offs

The physical world affords us very few free lunches. As a result, the principal goal of engineering is to understand the *trade-offs* inherent in any design scenario. In the case of consumer electronic devices like digital cameras, the trade-offs are stark and easy to understand: we want our cameras to be extremely small, have infinite battery life, deliver perfect quality results, and cost essentially nothing. We cannot achieve all of those goals simultaneously. Instead, we must understand the relative cost that improving the camera's performance on one goal will take on its other goals.

Let us understand some of the major goals for digital cameras in a little more detail. Realistic designs may take into account other goals as well, for example, durability and reliability. But this *design space* gives us good insight into why digital cameras look and operate the way they do. A complementary notion to design space is *use cases*—the scenarios under which we use a system. Different use cases often require systems carved from different parts of the design space.

Image Quality As we saw in Chap. 2, image quality is a complex topic, and we may judge different imaging systems as being of higher quality depending on what image characteristics we consider most important. But whatever we mean by image quality, some cameras simply do not need to produce very high-quality images—adequacy is more than sufficient in many use cases.

Physical Size Many use cases require small physical size. Overall physical size is an important goal. *Thinness* relative to the optical axis is particularly important in many consumer electronics use cases, smartphones being an obvious example. The size of the camera itself and the physical size of the user interface may be at odds. Smartphones can include very small cameras while still using a large display that is needed for other smartphone functions. Many professional cameras are physically large in part to provide multiple buttons and dials as well as a dedicated display.

Cost The manufacturing cost of a camera comes from several different sources: the cost of its components, assembly cost, software cost, and the cost of intellectual property licenses.

Power Consumption and Battery Life Digital cameras require electric power to operate; this power almost always comes from batteries. We want our batteries to be both long-lived and physically small. These two requirements are at odds thanks to

the most fundamental physical principles. Small, powerful batteries require high energy densities. The energy density of modern battery chemistries already approaches that of high explosives [Wol17].

3.3.2 Use Cases for Cameras

Use cases help us understand the requirements for systems. We can identify several common use cases for digital still and video cameras.

Still or Video Snapshots Snapshots are taken by non-expert photographers or perhaps knowledgeable photographers who do not want to worry too much about the process. These photos are often of people, though scenery may play a role as well. The photographer relies on the camera to deliver a usable picture or video. No additional lighting is provided; the available lighting may be poor.

Portraiture Portraiture is taken by experienced photographers in more controlled conditions. Portraits may be full-body or close-ups; they may include one or several people. The camera is typically within a few meters of the subject. Controlled environments and lighting are typical.

Landscape or Architectural Photography Photography or videography of natural scenes or buildings shares many characteristics with portraiture. However, the subject is usually much farther away from the camera, and the photographer has limited ability to control lighting.

Electronic News Gathering Electronic news gathering is videography performed under a wide range of conditions. The subjects may be people or events; the situations may be indoor or outdoor. The videographer is experienced but does not have much time to set up or monitor the camera.

Studio Videography Studio videography, such as for scripted television or cinema, may be taken indoors or outdoors. The videographer generally has more time for setup and monitoring. Several cameras may run simultaneously to capture the scene.

Technical and Scientific Technical and scientific uses for cameras range from laboratory studies to machine vision for manufacturing systems. These photographs may be taken either using a standard or specialized camera.

3.3.3 Four Examples of Camera Designs

Four different types of cameras cover the design space and use case space very well—webcams or surveillance cameras, smartphone cameras, mirrorless and SLR

cameras, and video camcorders. By comparing these four types of cameras, we can better understand the trade-offs inherent in digital camera design.

Webcams or Surveillance Cameras Webcams are accessories for computers; they may be built-in or plug-in accessories. Surveillance cameras stream video over a network to a server. They are designed to be small and inexpensive. Image quality is a secondary consideration. The webcam is powered by its host computer; the webcam's power consumption is a small fraction of the power consumption of most host computers. Webcams also operate under very different user interface assumptions than do our other categories. Much of the camera functionality is provided by software running on the host, not by software embedded in the webcam itself. This both reduces the cost of the camera and limits some aspects of its operation. Surveillance cameras require a little more computational support since they have to generate compressed video, but they also rely on the host for most device control operations.

The low-cost, small-size, and simplified operation of webcams all play together to determine its optical design. Webcams generally use simple lens systems with a small number of elements made from plastic. The lenses operate at small apertures to avoid the need for focusing—as a result, the lens has no moving elements. The lens is operated at a small focal length, which contributes to improving the parfocal distance and the range over which the camera stays in focus. Webcams have relatively small image sensors. As a result, the lens can have a short focal length, contributing to its small size.

The small aperture of the lens and image sensor both limit the webcam's image quality. As we will see in Section 3.sensor, small image sensors are more sensitive to noise, in particular electronic noise that is most visible at low illumination levels. The small aperture produces low illuminance levels, which make these noise sources more visible in output images.

Webcams are used for both still and video capture. Support for both is provided primarily by the host, as is storage.

Smartphone Cameras Smartphones are designed at a somewhat higher price point than are webcams, although they are still inexpensive relative to dedicated mirrorless and SLR cameras. They are physically very small, with thickness a particularly important design constraint. They are intended to produce higher-quality images than are most webcams and designed at a higher price point.

Smartphone optics are generally more capable than their webcam counterparts. Many provide focus and zoom. Given the small lens size, the elements can be moved by relatively simple mechanisms such as voice coils. These lenses are, however, simple relative to their mirrorless/SLR and camcorder counterparts.

Smartphone image sensors are very small. Their size is often determined more by depth than by width and height constraints. Figure 3.2 shows a cross section of a smartphone with its lens on one side and the image sensor on the other. The camera's width is approximately equal to the focal length of the lens. A 7 mm thick camera, for example, would allow a lens with a focal length of about 7 mm. If we assume a normal focal length lens, this gives an image sensor diagonal of 7 mm.

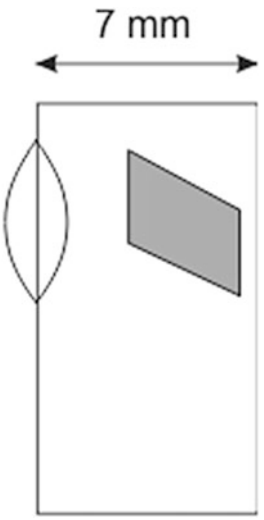


Fig. 3.2 Smartphone size constraints on camera design

Even a wide-angle lens equal to half of the normal focal length would give a 14 mm diagonal image sensor. As a result, they suffer from the same low-light noise limitations that constrain webcams.

Smartphone cameras are also part of larger devices, as with webcams, but smartphones are much more tightly integrated than are laptop and desktop computers. Smartphone processors provide specialized architectural features to support camera operations, including both still and video. Smartphone cameras are also constrained by limitations on battery capacity.

Mirrorless and SLR Cameras While these two categories are considered to be very different by the enthusiasts and professionals they target, they are quite similar under the hood, differing mainly in their viewing mechanisms. Mirrorless cameras use electronic viewfinders, while *single-lens reflex* (SLR) cameras use optical viewfinders.

Both are physically much larger and heavier than smartphone cameras. They rely on much larger image sensors and optical systems. As a result, they provide better operation under low-light operation. Their optical systems are complex, with lenses composed of many elements and powerful motors for focusing.

Camcorders Camcorders are optimized for video rather than still imagery. They are otherwise quite similar to mirrorless cameras. They may provide a wider range of video formats and settings than an SLR or mirrorless camera. They also provide features useful for videography. One example is *zebra stripes*, white stripes overlaid on the viewfinder image to identify overexposed areas of the frame. Many camcorders also provide neutral density filters to adjust exposure given that video provides fewer options for shutter speed than does still photography.

3.4 Image Sensors

This section studies image sensors in detail. Our goal is to use device physics and circuits to understand the imaging characteristics of sensors. After introducing some concepts in image sensor architectures, we analyze the characteristics of photosensors. We then study the two major types of silicon sensors: CCD and APS CMOS. We briefly consider advanced imagers such as time-of-flight and infrared sensors. Section 3.sensor.analysis uses these results to analyze the imaging characteristics of sensors. We close with a brief consideration of shutters, both electronic and mechanical.

3.4.1 Image Sensor Architectures

Image sensor size is one important parameter of an image sensor. Figure 3.3 shows the relative sizes of several common still image sensors. As you can see, these sensors range in size by over an order of magnitude.

Pixel size and *pixel count* are also important parameters; sensor size alone does not give us a great deal of information about the sensor's resolving power. Pixels generally range between 1 and 50 μm on a side. Pixel counts can vary widely from 1 to 100 megapixels. As we will see in Section 3.sensor.analysis, the resolution of a sensor depends on both its pixel size and its pixel count/image sensor size ratio. *Noise* is a key limitation on sensor performance; we will see that the most important sources of noise in image sensors are shot and reset noise.

The two major device designs for image sensors are the *charge-coupled device* (*CCD*) and the *CMOS* image sensor; we will discuss their design in more detail in Sections 3.sensor.ccd and 3.sensor.cmos. The CMOS image sensor now dominates many digital camera categories thanks to its low-cost manufacturing technology. However, CCDs are still used in some video cameras and have technical advantages that make them superior for high-performance applications.

Silicon is sensitive to all wavelengths of visible light, although it is most sensitive to red and infrared. We can sense color using color filters. By fabricating a color filter on top of each pixel, we can control the light that is allowed onto the pixel's photosensor. The most common pattern for color filter arrays is the *Bayer*

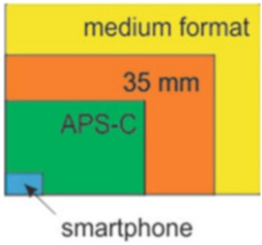


Fig. 3.3 Image sensor formats

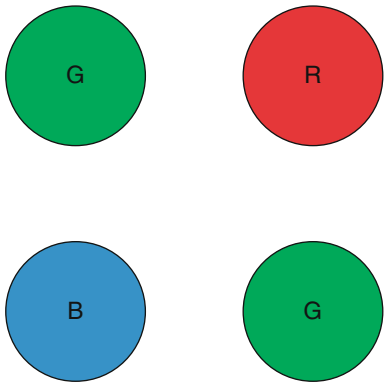


Fig. 3.4 The Bayer pattern

pattern [Bay75], shown in Fig. 3.4. The Bayer pattern forms a 2×2 pattern that is replicated across the image sensor. It uses two green filters to complete the pattern. The human eye is most sensitive to green, so the pair of green pixels can be used to approximate a luminance signal.

An alternative approach to color sensing was taken by the Foveon sensor [Lyo02]. It used stacked photodetectors: blue closest to the surface at a depth of 0.2 μm, then green at 0.6 μm, and then red at 2 μm. The stacked photosensors take advantage of the fact that the depth of penetration of light into silicon depends on wavelength. The color accuracy of the sensor depends on the depth accuracy of the fabrication of the photosensors.

Some camcorders use three CCDs, one for each primary color, and an image splitter to divide the incoming image to the separate image sensors. A few professional still cameras may also use a sensor without a color filter array and an external color filter wheel to successively capture red, green, and blue images.

Interlacing divides a video frame into *fields* with alternating lines—for example, all even lines in one field and all odd lines in the other. The fields are displayed sequentially. Interlacing was originally developed for early analog television systems to reach the flicker fusion rate at the lower frame rates possible at the time.

Many video image formats come from the US HDTV Grand Alliance specification: 1080p is 1920×1080 (columns x rows) in progressive format; 1080i is the same number of pixels but interlaced; and 720p is 1280×720 pixels in progressive format. These standards use the Rec. 709 color space. The UHD Alliance has defined ultrahigh-definition formats: 4 K UHDT is 3840×2160 pixels and is sometimes called *quad HD* and 8 K UHDTV is 7680×4320 pixels. These formats allow for both the Rec. 709 and Rec. 2020 color spaces to be used. The DCI standard measures slightly larger at 4096×2160 pixels.

Video is often encoded in luminance + chrominance format; YCrCb is one example. These formats may represent chrominance at lower spatial resolutions than is luminance. The *n:n:n* style is used to describe these formats: 4:4:4 samples both luminance and chrominance at full spatial resolution; 4:2:2 samples chrominance at half the spatial resolution of the luminance signal in both the horizontal

and vertical directions; and 4:2:0 alternates sending the two chroma signals on different frames, each at half the spatial resolution of luminance.

Some image sensors use *back-side illumination*. Traditional integrated circuits are fabricated with transistors at the top surface of the chip, with interconnection layers placed above. The back-side illumination scheme allows light in from the bottom of the chip to feed the photosensors on the opposite side. The chips are ground down to reduce the distance from the rear surface to the photosensor. Stacked sensors combine back-side illuminated sensors with other chips connected to the reverse side of the chip. The large number of connections and their relatively low parasitic impedances allow a variety of advanced features to be built, including high bandwidth and high frame rate sensing as well as attached processors.

3.4.2 Photosensors

Figure 3.5 shows the optical absorption coefficient α of silicon as a function of wavelength; the penetration depth is equal to the inverse of the absorption coefficient. Light in the red and infrared range penetrates most deeply into silicon. The dopants used to create n-type and p-type regions can be used to tweak the wavelengths to which the material is sensitive.

Sze analyzes photodetectors using the photoconductor [Sze81], which is a block of semiconductor with ohmic conducts at each end separated by length L. Photons absorbed by the semiconductor produce hole-electron pairs; the proportion of carriers generated by photons is known as *quantum efficiency* η. The primary photocurrent is

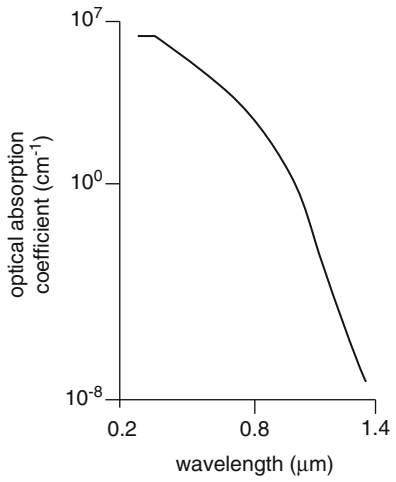


Fig. 3.5 Light absorption characteristics of silicon [Hon15]

$$I_p = q\left(\eta\frac{P_{\text{opt}}}{h\nu}\right)\left(\frac{\mu_n\tau\varepsilon}{L}\right) \tag{3.1}$$

where μ_n is the carrier mobility, τ is the carrier lifetime, P_{opt} is the optical power input to the device, and h is Planck's constant. The photocurrent gain is

$$A_{\text{opt}} = \frac{\tau}{t_r}, \tau = \frac{L}{v_d}. \tag{3.2}$$

t_r is the carrier transit time and the recombination rate is $1/\tau$. The response time depends on the transit time, which in turn depends on the distance over which the carrier must travel and the electric field which accelerates it.

The noise current in the photoconductor consists of three components: *thermal noise*, *shot noise*, and *1/f noise*. The mean-square thermal noise, also known as *Johnson noise*, due to the device conductance is [Sze81]

$$i_G^2 = 4kTGB \tag{3.3}$$

where B is the bandwidth of the device. The shot noise is proportional to τ/t_r.

The *dark current* is the current produced by the device when no illumination is applied. The bulk of dark current comes from recombination generation processes. A major source of dark current is traps in the silicon [Seq75]. This bulk recombination generation current is proportional to

$$\frac{1}{2}n_i\frac{x_d}{\tau_n} \tag{3.4}$$

where n_i is the intrinsic carrier concentration, x_d is the width of the depletion region, and τ_n is the minority carrier lifetime. Dark current from midband interface states can be analyzed in a similar way.

A *photodiode* uses a *pn* junction to collect photocurrent. A photodiode may be made as either a *pn* or *pin* (intrinsic) device. Photodiodes may be operated in either of two modes: *photovoltaic* mode operates at no voltage across the diode; *photoconductive* mode operates with a large reverse diode voltage. Shot noise dominates in photovoltaic operation. The photodiode capacitance is proportional to the Debye length and so is proportional to the inverse square root of the doping [Sze81].

$1/f$ noise occurs in many physical systems, including photodetectors, but its physical basis is poorly understood. In this case, f is the modulation frequency; $1/f$ noise is highest at low frequencies. Tian and El Gamal [Tia00] analyzed $1/f$ noise in APS sensors. Their analysis was based on a model for MOSFET $1/f$ noise based on the ability of gate oxide traps to capture channel carriers. This model predicts that $1/f$ noise is inversely proportional to gate area. They applied this noise model to the APS cell and found that the model predicts significantly higher noise values than does the traditional $1/f$ model. They also observed that $1/f$ noise in successive samples is highly correlated because the noise in each sample is generated by the same traps. As a result, double correlated sampling may actually increase $1/f$ noise.

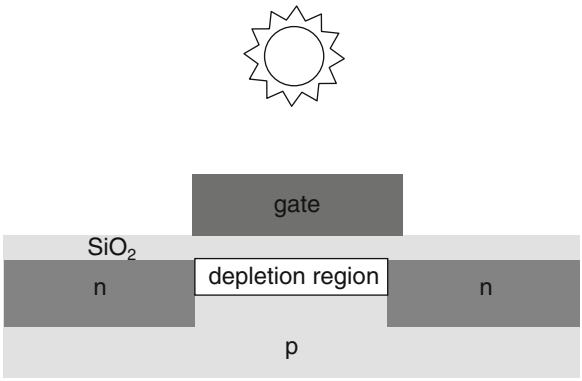


Fig. 3.6 Structure of a photogate

A *phototransistor* is a photodetector that uses transistor action to amplify the resulting signal. Both bipolar transistors and MOS structure can be used as phototransistors [Sze81]. In the case of a bipolar transistor, the photodetector is located at the base, causing a base current that is amplified in the emitter-collector current. A phototransistor based on an MOS capacitor is known as a *photogate* as shown in Fig. 3.6. The fact that the silicon gate absorbs a great deal of the blue spectrum limits the color response of the photogate. Photons pass through the gate and silicon dioxide into the device's channel region where they may be absorbed. The channel may be doped with various agents to adjust the wavelengths at which the device is sensitive. An applied gate voltage creates a depletion region in which minority carriers are collected. The gate voltage can be adjusted to change the depth of the channel's potential well relative to that of the source/drain region and selectively move the collected carriers to the source/drain. Some light is absorbed by the gate; windows may be cut into the gate to increase its transmissivity.

3.4.3 Charge-Coupled Devices

An image sensor is more than an array of photodetectors—we need to be able to read out the photodetector values quickly and accurately. The *charge-coupled device* (*CCD*) [Boy70] was the first practical solid-state image sensor.

The CCD is based on an array of MOS capacitors. The gate voltage (upper plate voltage) of the MOS capacitor can be used to control the depth of a potential well; with a high applied voltage, a deep potential well can be stored that can store a large number of electrons. The potential well depth is proportional to gate voltage, so we can manipulate collections of electrons using the relative gate voltages of adjacent MOS capacitors. An example three-terminal CCD [Seq75] is shown in Fig. 3.7. Each MOS capacitor is controlled by a clock phase φ_1, φ_2, and φ_3. When φ_2 is high, its potential well holds a collection of electrons. As we raise lower φ_2 and raise φ_3,

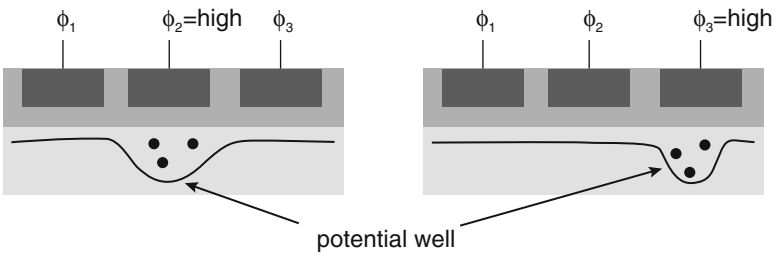


Fig. 3.7 Operation of a three-phase CCD

we can move electrons to the next MOS capacitor much as we might move marbles on a bedsheet. In this design, three terminals are used so that a barrier is introduced between each pair of active MOS capacitors.

3.4.4 APS CMOS Image Sensors

CCDs require specialized manufacturing processes. In contrast, the APS image sensor shares many similarities to dynamic RAM and can be made on processes similar to those used for DRAM. As a result, APS CMOS image sensors have come to dominate the market. CCD sensors are still used for applications where extremely low noise is required.

As shown in Fig. 3.8, an APS image sensor is organized much like a memory array but with continuous analog rather than discrete values. The pixel values are read a row at a time; horizontal signals provide row control. A column of pixels are connected to a *bit line*; circuitry at the end of each bit line reads the value from the bit line and transfers it to an analog shift register for readout. Capturing a frame proceeds in three steps:

- All pixels are reset simultaneously.
- All pixels are exposed simultaneously, with each integrating its own pixel illumination level.
- Pixels are read out a row at a time.

Figure 3.9 shows the schematic for an APS pixel cell based on a photodiode [Nix96]. A capacitance C_{pd} is used to integrate the photodiode current during the sample; this capacitor can be formed by a floating diffusion region. The reset transistor, when turned on, resets the capacitor voltage. The row transistor M_1 operates as a source follower to provide current gain. When the *row* signal is enabled, transistor M_{row} connects the amplified pixel value to the bit line.

Figure 3.10 shows the sample-and-hold circuitry in the column. The sample-and-hold circuit itself is formed by the sampling transistor M_{sample} and sampling

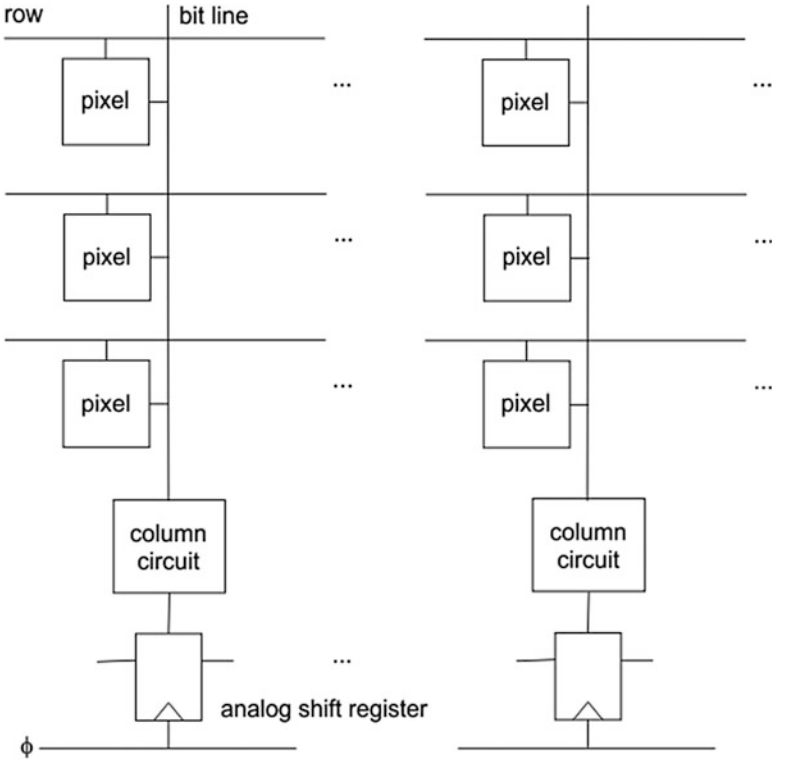
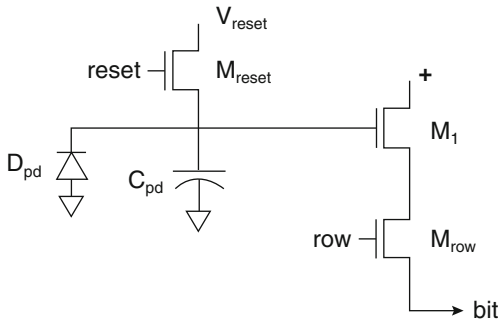


Fig. 3.8 Architecture of an APS image sensor

Fig. 3.9 Schematic of a photodiode-based APS pixel



capacitor C_s. M_{VLN} acts as a source follower for the column. To reduce noise in the circuit, a double sampling circuit is used to grab samples of both the reset value and the pixel value.

The sample-and-hold circuitry is connected to the remainder of the column readout circuits. Readout is designed to minimize two types of noise. *Fixed-pattern noise* results in a motley pattern of pixel values even when the sensor is not

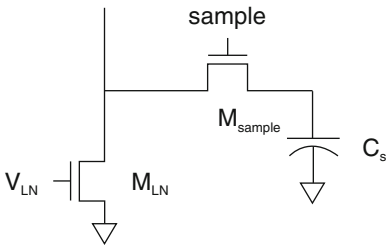


Fig. 3.10 Sample-and-hold circuits in the APS column

illuminated. This type of noise is the result of mismatches between the threshold voltages of the transistors on the bit line. The double sampling circuit is used to compare the reset and illuminated values of the pixel to cancel this noise. *Reset noise* is due to incomplete reset of the pixel. A full reset of the pixel takes approximately 1 *ms* largely because the reset transistor M_{reset} is in saturation for only part of the reset period [Zhe11]. Most applications do not allow for the reset period to be this long. As a result, the pixel is not fully reset; its value depends on the pixel value from the previous image. This phenomenon is known as *image lag*.

The column includes two sample-and-hold circuits: one for the integrated pixel value and one for the reset value [Nix96]. Differencing the pixel and reset values reduces image lag. After using the two values, the pixel and reset sampling capacitors are shorted together to produce a pair of output values. These output values are independent of the threshold voltage of the column driver transistors. As a result, this step eliminates column-based fixed-pattern noise.

Two groups analyzed noise in APS circuits, each emphasizing a different aspect. Yadid-Pecht et al. [Yad97] analyzed the sample-and-hold circuitry. They pointed out that the white noise power of an ideal sample-and-hold circuit is given by

$$\overline{v_n^2} = \frac{kT}{C_s}. \tag{3.5}$$

However, the gate-to-source capacitance of transistor M1 and the sensing node capacitance form a feedback network. Their analysis included both white and shot noise. Tian et al. [Tia99] noted that the photodiode capacitance depends on its reverse bias voltage. They analyzed the noise resulting from this nonlinearity as well as shot noise during reset. They analyzed mean-square reset noise voltage using a non-steady-state method; they found that this noise voltage is less than the value given by the traditional formula kT/C_{pd} where C_{pd} is the photodiode capacitance.

A standard pixel design has a dynamic range of 70 dB. We want larger dynamic range for both artistic and technical applications. Both the standard CCD and APS cells are linear; increasing the dynamic range of linear sensors may require a combination of higher operating voltages and larger pixel capacitances, both of which are undesirable. Improved circuit techniques allow pixels to operate nonlinearly and provide increased dynamic range; these sensors are often called *logarithmic sensors* since a logarithmic sensitivity curve matches the response of

the eye. Decker et al. [Dec98] used a varying voltage on the reset transistor to control the amount of charge kept in the pixel during the integration period, resulting in nonconstant charging rates. Consider the APS pixel circuit of Fig. 3-aps-pixel; if the reset voltage is varied during the integration time, the amount of charge kept on C_{pd} can be controlled resulting in a nonlinear response curve for the pixel. Schantz et al. [Sch00] used combined two techniques: each pixel could be accessed up to four times per frame, allowing for varying integration times; and column amplifier gains could also be controlled. Kavadis et al. [Kav00] used a separate reference current to allow for correlated double sampling calibration without requiring a separate measurement of the photocurrent. Stoppa et al. [Sto02] used a comparator to measure both low- and high-intensity levels with a comparator that determined when the photosensor charge had a reached a given level. Lee et al. [Lee06] developed an infrared image sensor that provided both high-dynamic range and high frame rates. Pixel values were recorded in floating-point format with a mantissa and exponent. Each pixel included a dual-slope ADC with separate comparators for the exponent and mantissa.

Dickinson et al. [Dic95] compared the quantum efficiency of APS and CMOS sensors. They found that APS sensors had reduced efficiency at wavelengths below about 500 *nm* but that their quantum efficiencies were otherwise comparable.

In addition to circuitry, image sensors include *color filter arrays* and *microlens arrays*. As shown in Fig. 3.11, the microlens helps to focus light onto the photosensor and helps to minimize the effects of fill factor. The color filter selects the wavelengths of light passed to the photosensor. The color of the filter can be controlled pixel by pixel, allowing, for example, the Bayer pattern filters to be built.

Most modern image sensors include their analog-to-digital converters (ADCs) on-chip. Some chips include several ADCs for higher performance. A variety of ADC architectures are used for image sensors: successive approximation, sigma-delta, and flash. An on-chip ADC allows the sensor interface to be digital. The MIPI D-PHY interface standard is commonly used to interface the image sensor to the rest of the camera system.

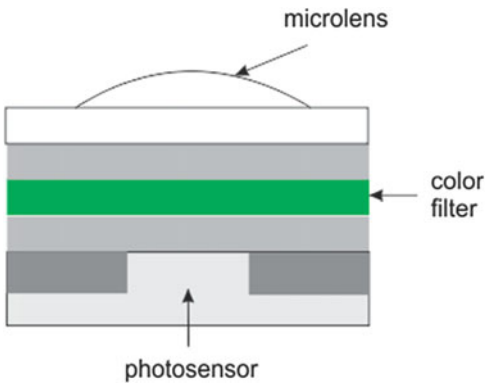


Fig. 3.11 Cross section of a pixel

Fill factor is the percentage of pixel area that is devoted to the photosensor.

Semiconductor image sensors are very linear—their response is proportional to the illumination [Whi74]. Some systems use *gamma* correction to adjust the response curve:

$$I_{\text{out}} = AV_{\text{in}}^{\gamma}. \tag{3.6}$$

3.4.5 Advanced Image Sensors

Kleinfelder et al. [Kle01] developed a high-speed image sensor that used an analog/digital converter per pixel. A pixel consisted of a photogate sensor, a comparator, and an 8-bit memory. The comparator compared the pixel value to a ramp voltage to generate the bits of the pixel value. They used correlated double sampling to cancel out comparator offset voltages. Their sensor was demonstrated to capture 352×288 images at 10,000 frames/sec. Some high-speed image sensors do not employ correlated double sampling to eliminate the settling time caused by the small voltage associated with the reset value. The lack of CDS results in increased pixel reset noise. Krymski et al. [Kry03] built a 240 frame/sec image sensor with an A/D converter per column. Xu et al. [Xu12] proposed connecting the photosensor to an integrating amplifier; the small capacitance of the amplifier would allow for faster settling times. Recent designs have integrated memory within the pixel as a burst buffer to allow for very high frame rates.

A *time-of-flight* sensor uses ranging techniques to measure the distance of objects from the image sensor. A laser pulse is sent to the subject and the pixel senses the time at which the reflected illumination returns from the subject. Time-of-flight sensing requires very fast electronic shutters to accurately measure the pulse return time. Elkhalili et al. [Elk04] designed a 4×64 sensor for time-of-flight measurements. Its shutter operated at 30 *ns*, and they pipelined sample acquisition with correlated double sampling to provide no dead time between measurements. The sensor could measure object distances up to 8 *m* at a 1 *cm* resolution.

The infrared band is very wide and different types of sensors are used for different parts of the band. Shortwave infrared is near the visible band. As we saw in Fig. 3.5, silicon is very sensitive to the shortwave infrared region. Image sensors designed for visible light are typically covered by a thin filter to absorb infrared radiation; we can use the image sensor for shortwave IR simply by stripping off the filter. However, this part of the infrared band requires illumination just as for visible light. *Thermographic images* are produced in the longwave infrared band—these are the types of images typically portrayed in movies; they rely on the heat produced by objects of interest, but they require specialized sensors. Two types of thermographic sensors are used [Fli12]: microbolometers and quantum-well infrared photon. Quantum-well sensors are more sensitive and faster but require that the sensor be cooled to cryogenic temperatures. Microbolometers,

in contrast, can be used at room temperature. Thermographic cameras also require optics made of different types of glass that do not absorb large amounts of long-wave IR.

3.4.6 Image Sensor Characteristics

An image sensor can be measured using three different metrics:

- *Pixel depth*, the number of bits used to represent a pixel value
- *Pixel count*, the number of pixels in the image
- *Pixel pitch*, the size of a pixel (which we assume for simplicity is equal to the distance between pixels)
- *Image sensor size*, the total physical size of the image sensor

These metrics are not entirely independent. We need to keep in mind that we quantize images both *spatially* and *intensity-wise:* we divide the image into pixels and then assign a discrete value to the intensity at each pixel. We will also see that the relationships between these metrics introduce trade-offs. In particular, making very small pixels to increase pixel count introduces some important limitations.

Pixel depth determines the number of luminance values that we can portray in the image. We represent the image, which has continuously varying intensity, as an integral number with discrete values. *Quantization noise* is the result of sampling the continuous image into discrete values. Our standard assumption is that image intensities are uniformly distributed over a range $[-Q/2, Q/2]$ where Q is the value corresponding to one bit. Then the root-mean-square (RMS) error is

$$E_q = \sqrt{\frac{1}{Q}\int_{-Q/2}^{Q/2} x^2 dq} = \frac{Q}{\sqrt{12}}. \tag{3.7}$$

Clearly, reducing the range covered by one bit reduces quantization noise. However, reducing quantization noise substantially below the physical noise in the imaging system yields no results. 10-bit image sensors yield a dynamic range similar to film; 12-bit and 14-bit dynamic range sensors are common advanced cameras and specific applications such as automotive.

However, small pixel depths can result in *posterization*. Using a small number of bits per pixel results in a small number of distinct values available in the image and clearly visible boundaries between regions with different values. This effect has its uses—this effect was commonly used in the 1960s, for example. But unwanted posterization can be distracting. We will see in Sect. 3.6.2 that compression algorithms can vary the number of bits per pixel throughout the image.

The *Kell factor* was introduced in the analog television era. It was motivated by the observation that images captured at the Nyquist limit for the subject appeared to have beat frequencies which could be minimized by limiting the bandwidth of the

image. Kell factor has been generalized for pixelated image sensors to refer to effective resolution. Diagonals through the sensor provide finer sampling than do the rows and columns of pixels.

To analyze the relationship between pixel count, pixel pitch, and image sensor size, let us assume that the image sensor is square and of size $s \times s$. If the pixel pitch is p, then the image sensor has $n = s/p$ pixels in each dimension, giving a total pixel count of n^2. We can use the same methodology we used for quantization noise to understand the RMS error introduced by pixelization:

$$E_p = \frac{p}{\sqrt{12}}. \tag{3.8}$$

Reducing pixel pitch reduces pixelization noise. Figure 3.12 shows an image sampled with both large and small pixels.

If pixelization noise were our only concern, we would want to make every image sensor with as many pixels as possible, independent of its physical size. However, we also have to consider the physical noise from the pixel. For simplicity, we will consider the reset noise of the imager (the noise generated due to incomplete resetting of the pixel value between frames). The RMS reset noise [Tia99] is

$$V_n = \sqrt{\frac{1}{2}\frac{kT}{C_{pd}}} \tag{3.9}$$

where C_{pd} is the photodiode capacitance, k is Boltzmann's constant, and T is temperature. Tian et al. give an example value of reset noise of 303 μV for some typical sensor values.

Pixel pitch is a proxy for photodiode capacitance. As a result, reducing the pitch increases reset noise. As a result, for any given image sensor size, we can find an optimal value for pixel pitch based on the competing mechanisms of electronic and pixelization noise, as shown in Fig. 3.13.

We can increase pixel count without reducing pixel pitch by increasing image sensor size. This effect takes advantage of the *capture-to-render ratio* or the relative sizes of the image sensor and the displayed image. Using a larger image sensor allows us to capture more pixels of a given size, which allows us to render each pixel as a smaller element in the rendered image. This relationship holds no matter what the final rendering size.

Photon shot noise is the most important source of noise in the photodetector. *Shot noise* is the result of the discrete nature of light—this is noise in the input signal, not in the image sensor device. Each photon registers as a shot; at large scales the quantization is not noticeable, but at the scale of pixels, we can see significant differences between the number of photons that hit pixels even when they are illuminated by the same object. The arrival of photons obeys the Poisson distribution; the arrival time of the next photon is independent of the arrival time of the last. The standard deviation is a measure of noise; the Poisson distribution relates mean and standard deviation as

Fig. 3.12 Pixelization and the effects of pixel pitch

$$\sigma_{\text{shot}} = \sqrt{\mu_{\text{shot}}}. \tag{3.10}$$

The signal-to-noise ratio of the light signal is

$$\left(\frac{S}{N}\right)_{\text{shot}} = \frac{\sigma_{\text{shot}}}{\sqrt{\mu_{\text{shot}}}} = \sqrt{\mu_{\text{shot}}}. \tag{3.11}$$

The signal-to-noise ratio of the photonic shot noise grows as the square root of the pixel illumination [The07]. We will return to this result below after we discuss exposure.

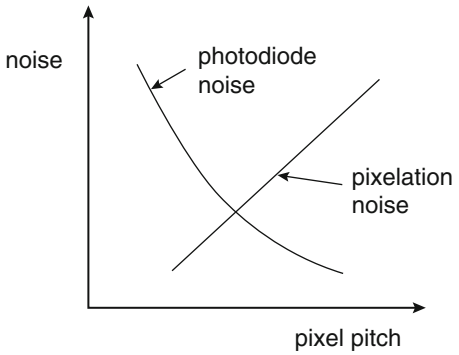


Fig. 3.13 Image noise vs. pixel pitch

Dark current is current produced in the absence of any illumination. The magnitude of the dark current is a principal limit in sensitivity. Several phenomena contribute to dark current [Tit11]: saturation current, generation-recombination current, direct tunneling, surface leakage, conduction through the oxide under large electric fields, and impact ionization.

Fixed-pattern noise results from spatial variations in the component parameters across the image sensor—gate capacitance, doping, etc. Fixed-pattern noise can be introduced at several points in the circuit, including the pixel amplifier, dark current sources, and column amplifiers; each has its own characteristics. Some amount of variation across the chip is both natural and inevitable; these variations cannot be fully eliminated from manufacturing. Once the fixed-pattern noise of a sensor is measured, it can be easily corrected by appropriate weighting of the pixels.

Image sensors also exhibit several other types of noise. The sample-and-hold circuit, for example, is a critical component that is subject to several types of noise. Gow et al. [Gow07] developed a detailed Matlab model of image sensor noise.

We also need to understand and measure the response of the photodetectors and circuitry to light. *Sensitometry* is the experimental evaluation of the response of an image sensor to light [Kod06]. We can measure that response by exposing the image sensor to light at a range of intensities and recording its response to provide a *characteristic curve* for the sensor. Figures 3.14 and 3.15 show example characteristic curves for film and image sensors, respectively. The characteristic curve is semilog: the x axis is the logarithm of exposure in units such as millilux-seconds. In the case of film, the y axis is the density of the image formed on the film. Film is a negative medium—higher exposure results in more silver and a darker image. The image sensor's characteristic curve has the opposite shape because it is a positive medium with higher exposure leading to higher *pixel values* or *pvalues*. We saw the effect of exposure on reference images in Fig. 2-changing-exposure.

The *dynamic range* of the image sensor is the ratio of the exposure values for maximum and minimum pixel values. The *contrast* of the image sensor is the slope of the characteristic curve. A typical film has a dynamic range of about ten stops. A standard model for the main part of the film characteristic curve is $\alpha + \beta^{\gamma}$. The film characteristic curve has lower slopes at both ends. These regions, called *toes*, have

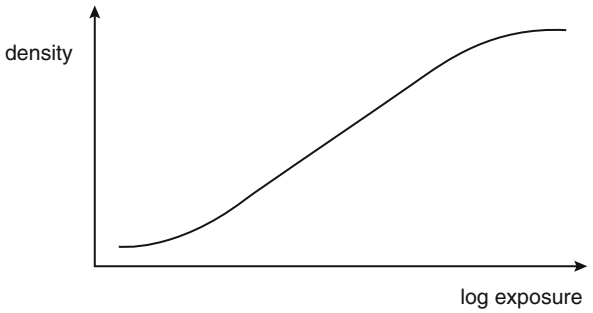


Fig. 3.14 A typical characteristic curve for film

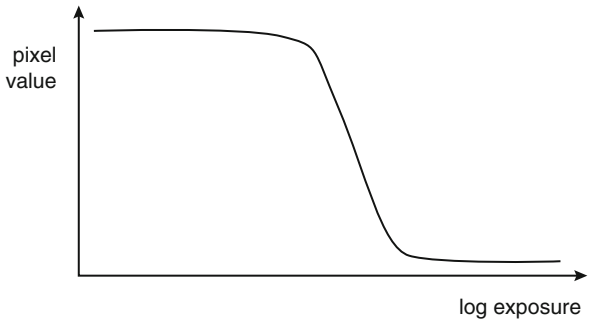


Fig. 3.15 A typical characteristic curve for an image sensor

lower contrast than the center of the characteristic curve. The toes of film give softer rendition of extreme light and dark regions and are part of the classic *film look*.

We measure the sensitivity of film by its *ISO number*, often referred to as its *speed*. The ISO number can be read from the characteristic curve. ISO 12232:2006 describes the standard for determining the ISO/ASA number for digital image sensors. ISO numbers are on a linear scale, so doubling the speed of the image sensor gives one additional stop of sensitivity. The ISO standard rounds speed numbers to standard values: 32, 64, 125, 250, etc.

An image sensor has a *native ISO* at which it provides typical responsiveness. Most image sensors and cameras allow the sensor to be used over a range of ISO values. The native response of the pixel values is multiplied by the *ISO multiplier* to create the adjusted ISO-valued image. We sometimes refer to non-native ISO values as either *push* (higher ISO) or *pull* (lower ISO) by analogy to film.

The effect of ISO multiplication is to multiply the pixel values; since ISO values are arranged at approximately powers of two, this scaling is equivalent to shifting the pixel values to the left. The scaling required for ISO multiplication can be performed at several points in the imaging chain as shown in Fig. 3.16. Early systems multiplied the signal value just before A/D conversion. However, when logarithmic sensors are used—the most efficient place to put ISO multiplication in the logarithmic control circuitry—shorter intervals for discharging the storage node result in a higher effective ISO. The reasoning behind this choice is illustrated in Fig. 3.16. A typical signal processing chain includes amplification and noise at each

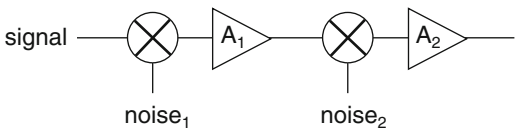


Fig. 3.16 Amplification of noise in signal processing chains

stage. The first stage amplifies both the signal and its noise, which tends to cause the first stage's noise to dominate over that of later stages.

Attempts to translate the Zone System into digital photography have resulted in some confusion and inaccurate information. Some authors claim, either explicitly or implicitly, that changing the exposure of an image changes the slope of the characteristic curve. Nothing could be further from the truth. The characteristics of the pixel circuits (and to some extent the surrounding circuits) determine how a given number of photons are translated into a pixel value. Nor do the higher bit positions somehow carry more information—pixels obey the laws of arithmetic and a bit is a bit. Think of a bit as a unit of just noticeable difference—if we increase a pixel's value from 200 to 201, we have increased its intensity by the same amount as if we increased it from 5 to 6.

The common digital interpretation of the Zone System is the *expose-to-the-right* rule—that exposures should be as high as possible (pushing the histogram to the right) without clipping highlights. This rule does provide benefit but not for the reasons commonly believed. For a given ISO value, this rule has some value as it pushes the signal further above the sensor's read noise. When used with ISO multiplication, it has an even greater benefit. As we saw above, photon shot noise grows with the square root of the signal. If we use the logarithmic sensor signals to control the sensor ISO (as compared to performing an amplification or digital multiplication at the end of the signal processing chain), then the higher pixel value is propagated through the image sensor chain. A higher illumination value gives us a larger spread between signal and shot noise thanks to the Poisson distribution characteristics. Because noise through a chain of amplifiers is dominated by the first stage's noise, we improve the signal-to-noise ratio through the entire chain. The result—one that is surprising from the point of view of film photography—is that higher ISO values result in lower noise. We will return to the Zone System in Section 4.tonalmapping.

Some digital cameras exhibit *ISO invariance* [Say15]: the noise in the image is roughly independent of the ISO setting. If the image sensor has a wide dynamic range relative to the scene and a low analog noise floor, then we do not need to apply the expose-to-the-right rule when the photo is taken. Instead, we can move the response curve to higher levels in post-processing, a process known as *pushing*, thus increasing the levels of the shadows. Tonal mapping using ISO invariance has the advantage of not increasing the exposure of the highlights, thereby reducing the chance of saturating those highlight regions.

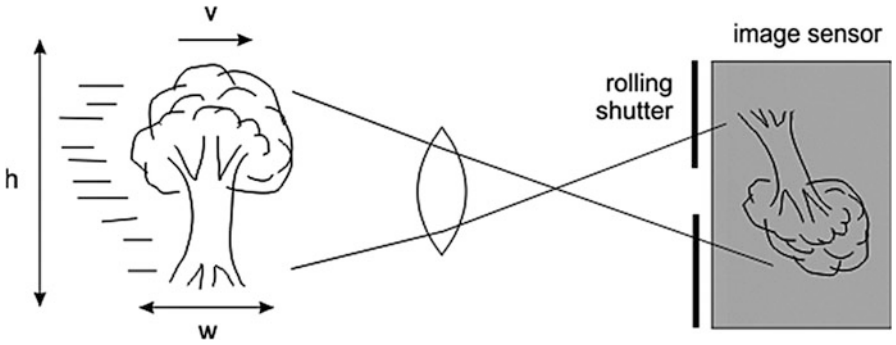


Fig. 3.17 Moving subjects and rolling shutters

Motion blur is a form of temporal sampling noise. As the subject moves during exposure, the pixels integrate light during the exposure interval e not from a single point on the subject but along a track:

$$M(e) = \int_0^e I(t)dt. \tag{3.12}$$

Capturing motion with a rolling shutter results in tearing or angling of the moving subject. This phenomenon entered the visual vocabulary as a depiction of speed but is entirely due to the effects of sampling by moving shutters. (This phenomenon was first captured by film cameras of live subjects; it then became a trope in cartoons.) Consider the moving subject in Fig. 3.17. The subject is of height h, width w, and traveling at a velocity v. For a given shutter speed s, the subject moves a distance $l = vs$ during the exposure. At a speed of $v = w/s$, the subject moves its entire width in one exposure interval; tearing of even 10% is noticeable.

Many advanced cameras provide video capture but have sensors whose resolution is considerably larger than that required for the supported video formats. These cameras generally subsample lines to match the required video resolution. This process means that parts of the image are not sampled at all, a situation very different from the subpixel motion estimation we will discuss in Section 3.h264.

3.4.7 Shutters and Irises

Cameras need shutters to control exposure. Several types of shutters can be used, each with their own advantages. *Electronic shutters* such as that of Reich et al. [Rei93] use diffusion regions to selectively sweep electrons away from the photodetector when the appropriate voltage is applied to the diffusion. Electronic shutters are widely used because they eliminate the size and expense of a mechanical shutter; they also respond very quickly. However, electronic shutters are not as

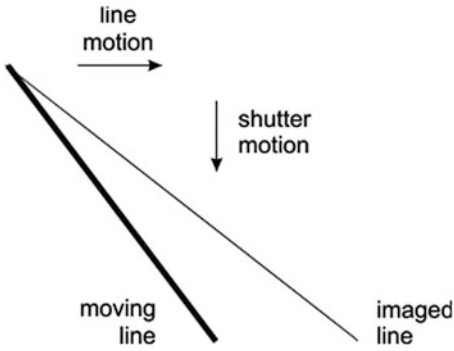


Fig. 3.18 Aliasing during rolling shutter operation

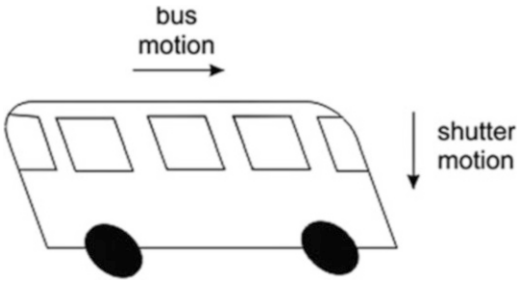


Fig. 3.19 Leaning of moving objects due to shutter motion

effective as mechanical shutters in cutting out light—electrons that are not captured by the shutter structure result in noise in the final image. Some advanced cameras use both mechanical and electronic shutters. Typically, the mechanical shutter is used by default, but the electronic shutter can be used to eliminate the vibration caused by the mechanical shutter.

An electronic shutter may be operated as either a *global shutter* or a *rolling shutter*. A global shutter opens and closes all pixels simultaneously. A rolling shutter, in contrast, shuts off pixels a line at a time. Rolling shutters can cause aliasing artifacts with moving images as shown in Fig. 3.18.

Leaf shutters are placed within the lens and use interlocking vanes. *Focal plane shutters* are located in front of the image surface and move a curtain across the image surface. Leaf shutters generally provide higher flash synchronization speeds. As shown in Fig. 3.19, focal plane shutters can result in fast-moving object leaning—the curtain opening scans the moving object to capture it at different locations in different parts of the image. This physical effect has become a visual symbol for motion in still photos.

Cinema shutters must be able to repeatedly expose images. Video cameras may use either electronic shutters or rotating shutters. Video necessarily provides fewer options for shutter speeds than are possible with still photography; many video cameras provide neutral density filters to adjust exposure. Mechanical shutters for cinema are sometimes described by the angle through which they expose the image sensor. Figure 3.20 shows an example of a 270° shutter.

Fig. 3.20 A 270° shutter

A shutter gives one parameter for exposure variation. Irises provide a second parameter and provide reciprocity as discussed in Sect. 2.8.1. However, many smartphones do not include irises. They instead use ISO scaling to provide a second exposure parameter.

3.5 Preexposure Operations

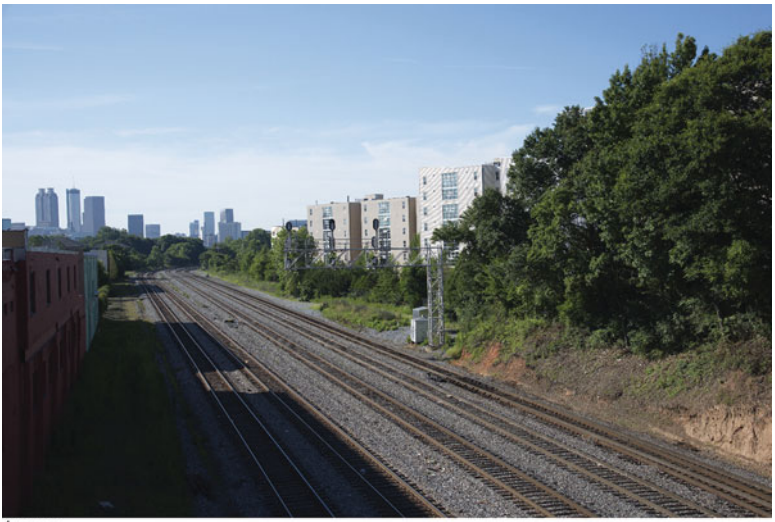
Before capturing an image, the camera must determine its focus and exposure. These steps are critical to previsualization of the image. As we saw in Sect. 2.8, the subject of the image may not be obvious, making it harder to determine how we should focus and expose. Algorithms for focus and exposure necessarily have a heuristic element to take into account the varying goals of a photograph. We need to solve two distinct problems for both focus and exposure: how do we determine focus/exposure at a given point in the image and which point do we choose to evaluate. Since many photographs are of people, *face detection* provides an important clue for both focus and exposure.

Our focus and exposure algorithms must be fast. Autofocus may be used on fast-moving subjects; exposure can also change quickly as subjects move and lighting changes. Both operations must be performed at rates of fractions of a second.

Many (but not all) of these exposure and focus algorithms make use of data from the image sensor. In some cases, particularly in SLRs, a beam splitter may be used to send some of the light to a separate image sensor for measurement purposes. However, most digital cameras read from the image sensor at reduced resolutions.

An important abstraction of the image that plays an important role in preexposure is the *histogram*. We can make a histogram of pixel values in any of several representations: luminance, RGB, etc. Figure 3.21 shows an image along with its luminance and RGB histogram. The histogram is divided into a set of *bins* that represent a range of pixel values. For each pixel, we increment the count in the bin that represents that pixel's value. The shape of the histogram tells us a surprising amount about the image in a very compact representation.

We will separately discuss focus in Sect. 3.5.1 and exposure in Sect. 3.5.2. Section 3.5.3 considers image stabilization. We will devote Section 3.5.4 to face



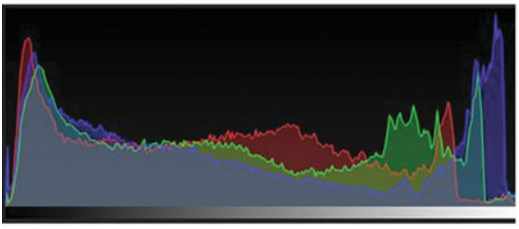


Fig. 3.21 Histograms of an image

detection, which is important to both focus and exposure. Focus and exposure are similar for video and still photography; the main differences lie in whether they operate continuously or remain fixed for the duration of the shot.

3.5.1 Autofocus

Autofocus systems predate digital cameras by several decades. Early autofocus systems generally used *active* autofocus because they did not have an image sensor with useful resolution. Active autofocus systems use pulses to determine the range

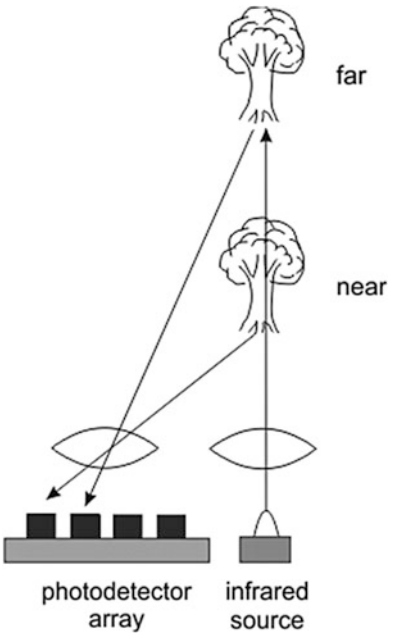


Fig. 3.22 Infrared autofocus by triangulation

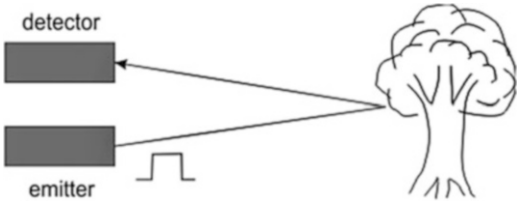


Fig. 3.23 Pulse reflection and time-of-flight ranging

of the subject. Infrared, ultrasound, and visible light pulses can be used, but infrared systems operate under fundamentally different techniques than do ultrasound or visible light ranging systems.

Infrared autofocus systems operate by *triangulation* as shown in Fig. 3.22. (Some cameras and flash systems also use IR pulses to illuminate the subject for passive autofocus systems; those systems do not rely on triangulation.)

Ultrasound and visible light time-of-flight systems both operate by measuring the time from the emission of a pulse to the detection of a reflected return signal as shown in Fig. 3.23. Radar operates on a similar principle although its pulses are in the radio band. However, given the vastly different speeds of sound and light, the electronics required for these two methods are vastly different. The speed of sound is roughly 343 m/s, while the speed of light is 3×10^8 m/s. Ultrasound detectors can use straightforward timing circuits (typically by charging a capacitor) to measure time-of-flight. Visible light time-of-flight sensors, as we discussed in Section 3. sensor.advanced, require 30 *ns* timing. Some versions of the Polaroid SX-70

[Lan72] offered an early autofocus system using ultrasound; its sensor system interface exposed two signals, one of which carried the timing of the outgoing pulse, with the other giving the pulse timing for the return signal.

All these active autofocus systems operate only under a limited range subject distance—the subject must be close enough to reflect a signal large enough to be detected. If the system does not detect a return signal, it can default to focus at infinity. Some modern systems combine optical time-of-flight with a passive autofocus system.

Passive autofocus systems are widely used. The two major approaches are *phase detection* and *contrast detection*. Contrast detection is widely used in non-SLR cameras because it makes direct use of the image sensor. Phase detection is primarily used in SLRs because it requires additional optical mechanisms (although on-chip phase detection sensors have recently appeared). Phase detection is faster than contrast detection, offsetting its increased hardware complexity for high-end cameras. Both methods make use of local features such as lines—we cannot focus on a completely undifferentiated, featureless surface.

As shown in Fig. 3.24, phase autofocus systems [Sta76, Yam81] take advantage of the fact that rays from a point on the subject may enter the lens at many different points, all of which are focused at the same point in the focal plane. We can be out of

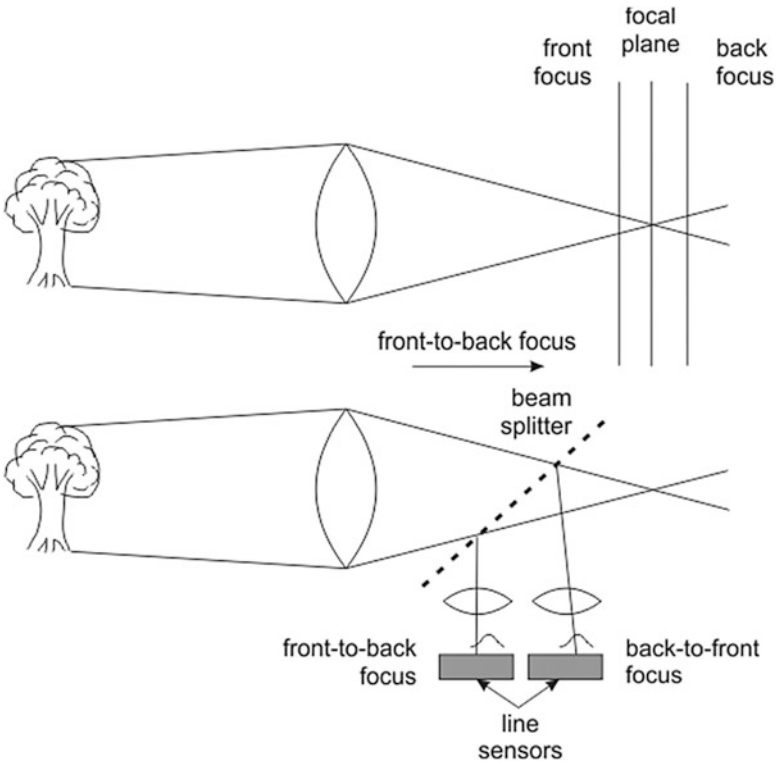


Fig. 3.24 Phase autofocus

focus in two different ways: *front focus* for a lens position that puts the image sensor too close to the subject and *back focus* when the image sensor is too far away from the subject. The phase autofocus system picks off the incoming from two conjugate points; it does so to allow for separation of the sensors for the two different versions of the image. The light can be directed to the autofocus system by, for example, a beam splitter. A pair of lenses focuses each of the test images onto its own line sensor that sense intensity at several points along a line in one dimension. The line sensors give an intensity profile for each of the test images. Those profiles can be compared in shape to determine the relative offset of the two test images. When the subject is in focus, the two test images will have a known offset. The offset between the test images gives both the magnitude and direction of the focus action. In the figure, if the subject is in front focus, the test images will be farther apart, while if in back focus, they will be closer together. Phase autofocus is fast because it can determine which direction to drive the lens for focus. However, phase autofocus is sensitive to orientation—the line that provides the feature must be perpendicular to the line sensors. Phase autofocus can be integrated onto the image sensor with dedicated pixels. One approach adds masks to the microlenses of these pixels that ensure that only light from the required direction are allowed into the phase detection pixels [But10]. Another approach uses aspherical lenses to direct focus to one side.

Contrast detection systems [Bel92] measure the contrast between adjacent pixels to determine focus. Contrast at an edge is highest when the edge is in focus. Unlike phase detection, contrast detection does not require special or modified hardware and can be performed directly on pixel values read from the image sensor. As a result, contrast autofocus is well-suited to video since it does not require additional hardware in the optical path, and focusing decision can be made from pixel data read from the sensor. However, contrast detection does not directly indicate whether the subject is in front focus or back focus. Once the focus starts to move, the size of the blur circle will decrease if the subject is coming into focus and increase if the subject is going out of focus. However, the initial choice of direction for focusing is indeterminate. The hunting required to find the focus point makes contrast detection slower than phase detection. Several algorithms can be used to evaluate focus [Che01]:

- *Sum-modulus-difference* forms the sum of difference of adjacent pixels.
- *Histogram entropy* is defined as $-\sum_{h(i)\neq 0} h(i) \ln h(i)$ for the bins of the histogram $h(i)$.
- *Histogram of local variations* finds the best-fit line through the logarithms of the histogram bins.
- *Fast Fourier transform* evaluates the FFT of the image region.

Contrast autofocus is also orientation-dependent. However, the orientation of the contrast measurement can be changed more easily by proper pixel readout and arithmetic since the measurement does not rely on specialized hardware.

Autofocus must be performed at a particular point in the image. Phase detection requires specialized hardware at the autofocus points; contrast detection may limit

Fig. 3.25 Focus point placement in the image region

itself to certain points to simplify the algorithms. The autofocus system must determine which points to use for focus. Both phase and contrast algorithms rely on line-like features to provide something on which to focus as shown in Fig. 3.25. Given that the autofocus points are fixed and relatively sparse, only a few points will be aligned with a useful focusing feature at any given framing. When more than one focus point is available and the points have different focus values, heuristics may be used to select a focus point, such as preferring ones toward the center of the image. Advanced cameras typically allow the user to select an autofocus point. Some autofocus systems perform motion estimation at the autofocus points to track the subject and move the focus point as appropriate. Some autofocus systems also use motion estimation to predict the required change in focus for a fast-moving object in order to take into account the lag from the final focus measurement to the actual image exposure.

3.5.2 Exposure

Autoexposure systems also predate digital cameras. Built-in meters produce reflected exposure values (unless the photographer turns the camera around to capture the light onto the subject). Built-in light meters first appeared in the 1960s. The earliest meters did not make use of the imaging path optics; later cameras introduced *through the lens (TTL) metering*. These cameras did not have motors to drive the iris and shutter selectors, so the photographer turned these selectors to, for example, center a needle.

The first light metering systems took a single reading of the entire scene, providing an averaged reflective reading. The *center-weighted* system introduced by Nikon in the 1960s placed additional weight on a circle covering the middle part

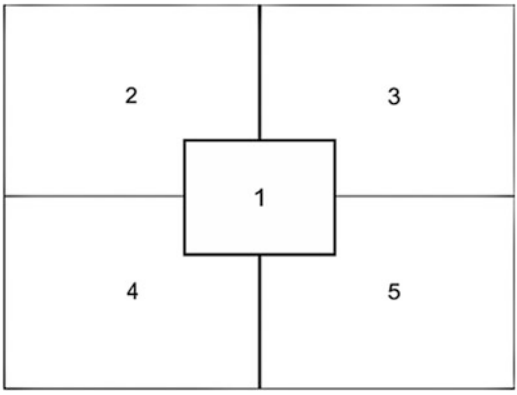


Fig. 3.26 Example exposure zones

of the frame under the assumption that the subject was likely to appear in that circle. The next move was to *exposure zones*. Figure 3.26 shows a typical configuration of zones, with one in the middle and other zones surrounding the center. After taking a reading in each zone, the camera can apply rules to determine an overall exposure.

Film cameras relied on separate exposure sensors; manufacturing cost limited the number of sensors that could be put on the camera. Digital cameras can take exposure readings directly from the image sensor and have much greater freedom in using and combining the pixel values to determine an exposure. However, the principle of measuring exposure at different points and then interpreting those exposure values still holds. For example, the exposure system can test a given exposure to be sure it falls within an acceptable range $[EV_{\min}, EV_{\max}]$; if not, it can try to determine exposure from a different set of points [Tsu93]. Consider, for example, a photo of a person standing in front of a bright window. The camera first determines an exposure based on zones 2 and 3 in Fig. 3.26 and then determines that the exposure value is too high. It can then try to compute an exposure value based on zones 4 and 5. If that value is unacceptable, it can try an exposure based on Zone I. Some cameras allow the user to set a scene type, such as landscape or portrait. The scene type can be used to determine the exposure points to be used.

We can use the luminance histogram to determine exposure [Bel02]. We can test for clipping by determining whether a given percentage of pixels (perhaps 5%) are congregated at the ends of the histogram. Testing for the position of the histogram center is a secondary test; we prefer the histogram to be in the middle or perhaps slightly toward the top.

Video cameras want to avoid breathing of the exposure caused by sudden changes in light. The exposure is adjusted continuously during shooting. Using a lower gain response at high exposure levels than for lower exposure levels avoids causing large changes in exposure when a small, bright region comes into the image [Kon92].

Once we know the exposure value for the image, we still need to determine the shutter speed and aperture. As Fig. 3.27 illustrates, reciprocity gives us *equal*

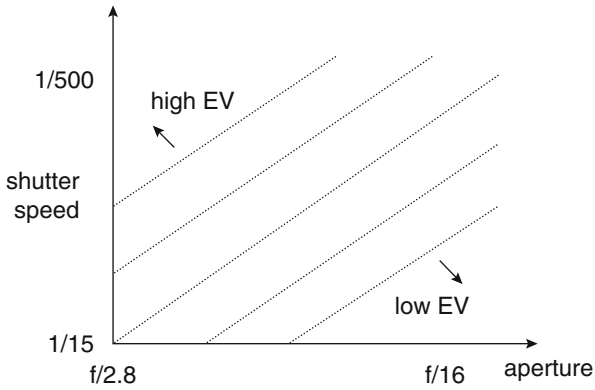


Fig. 3.27 Equal exposure lines

exposure lines for each EV. (The term *isoexposure line* would be nice, but that name risks confusion with *auto ISO*.) Cameras generally avoid low exposure speeds, typically using shutter speeds of 1/125 sec or higher to avoid camera shake. Once the minimum shutter speed has been satisfied, the camera can start to reduce the aperture to increase depth-of-field.

The image sensor has a native ISO or sensitivity. Most image sensors place amplifiers in the imaging chain that can be used to amplify the pixel values coming off the sensor. This amplification effectively increases the ISO of the sensor [Par97]. The camera's exposure heuristics can, when in *auto ISO mode*, choose to increase the effective ISO rather than fall into an unacceptably low shutter speed. However, ISO compensation amplifiers also amplify sensor noise that can affect image quality.

3.5.3 Image Stabilization

Image stabilization has different uses in still photography and video. We use it with for still imagery in large part to reduce the minimum shutter speed required for handheld photos or, equivalently, to use a smaller aperture at a given shutter speed. In contrast, we stabilize video sequences to reduce or eliminate the jitter visible in the shot.

We can stabilize the image against shake using several different methods: *optical image stabilization* (*OIS*) moves the optics; *mechanical image stabilization* moves the image sensor; and digital stabilization performs image processing. Camera movement can be determined either optically or through sensors such as accelerometers.

Optical image stabilization senses motion and moves optical elements to change the optical path to compensate. Figure 3.28 [Oiz93] shows a pair of a negative and a positive lens; together they provide little or no magnification. However, shifting the negative lens perpendicular to the optical axis will adjust the focus points to compensate for the motion of the camera.

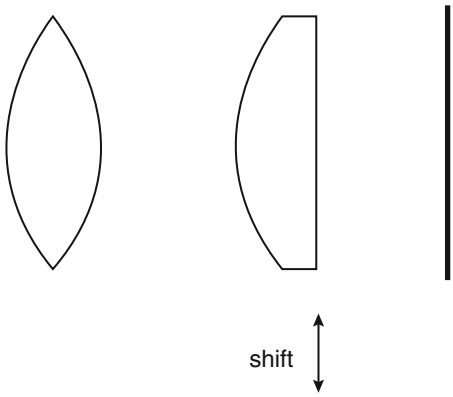


Fig. 3.28 Optical image stabilization [Oiz93]

Mechanical image stabilization senses camera motion and moves the image sensor to accommodate. The image sensor can be moved using piezoelectric actuators. The piezoelectric effect relates mechanical and electrical energy in crystals; it can be used to generate precise motion.

Still image stabilization can be used to counteract both blurring and geometric distortions of the subject. One approach [Heb08] takes advantage of a rolling shutter to divide the image into horizontal strips. Camera motion is determined either by image analysis or using accelerometers. A deskewing transformation is created for each strip based upon the camera motion. Each strip is deblurred, the deskewing operation is applied, and then the image is reformed from its component strips.

We will discuss video stabilization in Chap. 4.11.2.

3.5.4 Face Detection and Tracking

First, we need to clearly distinguish between *face detection* and *face identification*. The second identifies the face of a particular person at a given location in the image; the first only identifies a generic human face. Face identification is important in a number of applications but is not particularly useful for focus or exposure determination. Merely knowing the location of the face or faces in an image is more than sufficient to determine where focus and exposure algorithms should be applied.

Several approaches to face detection and recognition have been developed [Yan02]. Facial detection often combines image features extracted bottom-up with models of the organization of the typical human face. Figure 3.29 illustrates the types of facial features that can be used for recognition, based on a cartoon-style description of the face: the eyes, nose, mouth, etc.

Many fast face detectors are based on the approach of Viola and Jones [Vio01]; they combined simple features to rapidly prune the search space. As classifiers, they used windows known as *Haar-like features*. Figure 3.30 shows the features of size 2, 3, and 4. The 2-pixel configuration gives two different classifiers: $a_1 - a_2$ and

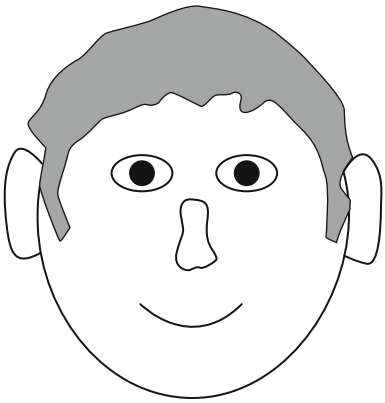


Fig. 3.29 Features in a typical face

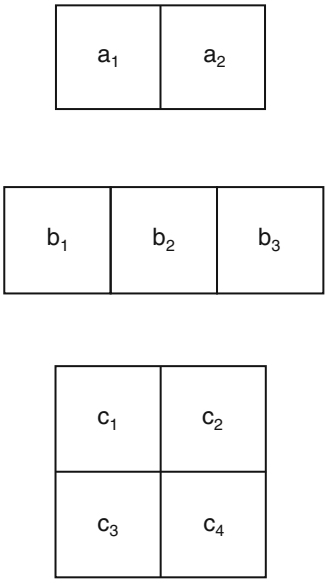


Fig. 3.30 Haar-like features

$a_2 - a_1$. The size three window classifier computes $b_1 + b_3 - b_2$. The size four window classifiers are $c_1 + c_3 - c_2 - c_4$ and $c_2 + c_4 - c_1 - c_3$. These classifiers can be combined into larger windows; Viola and Jones used a window of 24×24 which contains over 180,000 windows. Lienhardt and Maydt [Lie02] expanded the set of features to include rotated rectangles.

They use the *integral image*, illustrated in Fig. 3.31, as an intermediate representation for fast computation of features in different parts of the image. The integral image for point p_4 is the sum of all the pixels above and to the left, including the point itself:

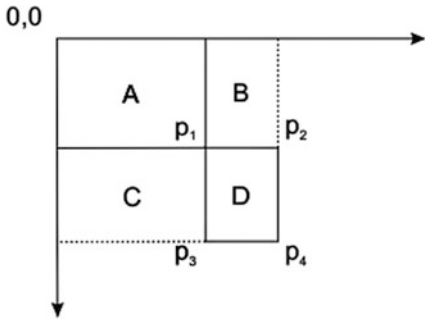


Fig. 3.31 The integral image of a point and its decomposition

Initialize weights: $w_{1,i} = \frac{1}{2m}$ if $y_i = 0$, $w_{1,i} = \frac{1}{2l}$ if $y_i = 1$

For $t = 1, \cdots, t$:

Normalize weights $w_{t,i} = \frac{w_{t,i}}{\sum_{1 \le j \le n} w_{t,j}}$.

For each feature j, train single-feature classifier h_j. Evaluate error $\epsilon_j = \sum_j w_i |h_j(x_i) - y_i|$.

Choose classifier with lowest error.

Update weights $w_{t+1,i} = w_{t,i}\beta_t^{1-\epsilon_i}$

where $e_i = 0$ if x_i is classified correctly

and $e_i = 1$ if classified incorrectly, $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

Result: $h(x) = 1$ if $\sum_{1 \le t \le T} \alpha_t h_t(x) \ge \frac{1}{2}\sum_{1 \le t \le T} \alpha_t$, $h(x) = 0$ otherwise, $\alpha_t = \log \frac{1}{\beta_t}$.

Fig. 3.32 The AdaBoost algorithm

$$II(p_{x,y}) = \sum_{i \le x} \sum_{j \le y} I(i,j). \tag{3.13}$$

We can compute the value of regions that do not extend to the origin as combinations of integral images—for example, $D = II(p_4) - II(p_1) - II(p_2) - II(p_3)$.

Viola and Jones used the AdaBoost algorithm, shown in Fig. 3.32, to train classifiers. The algorithm is given a set of n training images $\{x_i, \cdots, x_n\}$; for each one, we have a training value $y_i \in \{0, 1\}$ to identify negative/positive results. The training set has m negative results and l positive results. They combined the classifiers into a *cascade* designed to winnow out unpromising subwindows. The cascade is fed all subwindows; only subwindows that pass the first classifier are passed to the second and so on. Cho et al. [Cho09] developed a hardware face detector based on the approach of Viola and Jones.

Theocharides et al. [The04] developed a hardware face detector that was invariant to rotation. Their architecture generated an image pyramid, then performed rotation including lighting correction, and was then classified using three parallel neural networks.

3.6 Postexposure Operations

Postexposure processing has two important goals. First, it generates a complete, usable image. Color filter array interpolation creates full RGB values for every pixel; white balance adjusts for the color temperature of the scene's lighting. Second, postexposure processing improves the image. Sharpening produces results that please viewers.

Several of these algorithms require digital image filtering, so we will introduce some concepts and notations now. Filtering of images is a two-dimensional form of digital filtering. Each pixel of the result R is a function of the values of some of the pixels of the source image I:

$$R(i,j) = \sum_i \sum_j f(I). \tag{3.14}$$

Images are traditionally placed in the fourth quadrant; this practice began with analog television, which scanned the screen starting from the upper-left.

Many, though not all, operations are *linear* and can be described as a combination of the source image pixels multiplied by coefficients:

$$R(i,j) = \sum_i \sum_j c(i,j)I(i,j). \tag{3.15}$$

We will use i for rows (x) and j for columns (y). We refer to the range over which the filter operates as its *window*. We can specify the filter coefficients as a window whose indexes are relative to the center of the window as shown in Fig. 3.33.

3.6.1 Color Filter Array Interpolation

The image sensor's color filter array gives us a pixel value of a particular color at each location. We want to have a full-color pixel value—such as RGB—at each point. *Color filter array interpolation* fills in the missing color components at each pixel. These algorithms are also known as *demosaicing*, but the term *mosaicing* is also used for other image processing operations.

We can understand CFA interpolation using the Bayer pattern; the same approach can be applied to other filter patterns as well. As shown in Fig. 3.34, a green pixel is horizontally or vertically adjacent to two red and two blue pixels.

c(1,-1)	c(1,0)	c(1,1)
c(0,-1)	c(0,0)	c(0,1)
c(-1,-1)	c(-1,0)	c(-1,1)

Fig. 3.33 A filter coefficient window

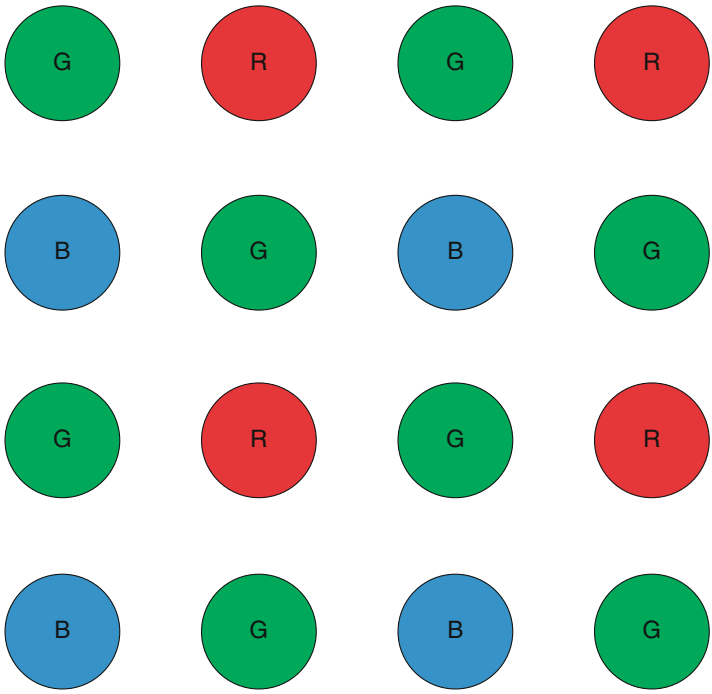


Fig. 3.34 Color filter array interpolation

Each red or blue pixel is adjacent to four green pixels but is only diagonally adjacent to pixels of the complementary color.

A very simple approach is *bilinear interpolation*. We can find the missing color components for a green pixel as

$$R_G(i,j) = \frac{1}{2}[I(i,j-1) + I(i,j+1)] \tag{3.16}$$

$$B_G(i,j) = \frac{1}{2}[I(i-1,j) + I(i+1)]. \tag{3.17}$$

We can find the missing values for a blue pixel as

$$G_B(i,j) = \frac{1}{4}[I(i,j-1) + I(i,j+1) + I(i-1,j) + I(i+1,j)]. \tag{3.18}$$

$$R_B(i,j) = \frac{1}{4}[I(i-1,j-1) + I(i-1,j+1) + I(i+1,j-1) + I(i+1,j+1)]. \tag{3.19}$$

And similarly for the red pixel

$$G_R(i,j) = \frac{1}{4}[I(i,j-1) + I(i,j+1) + I(i-1,j) + I(i+1,j)]. \tag{3.20}$$

$$B_R(i,j) = \frac{1}{4}[I(i-1,j-1) + I(i-1,j+1) + I(i+1,j-1) + I(i+1,j+1)]. \tag{3.21}$$

We can also generate use *bicubic interpolation* methods that average pixels over a larger area.

However, the simple filtering approaches result in image artifacts at edges; the effect is clearest at a boundary between white and a darker background. As shown in Fig. 3.35, each of the color interpolation filters estimates that the white/dark boundary is at a slightly different position. They do so because they are offset from each other. As a result, the white/dark line is rendered as three distinct lines, one for each color.

A variety of more sophisticated CFA interpolation algorithms have been developed [Gun05]. *Edge-directed interpolation* tries to identify horizontal or vertical

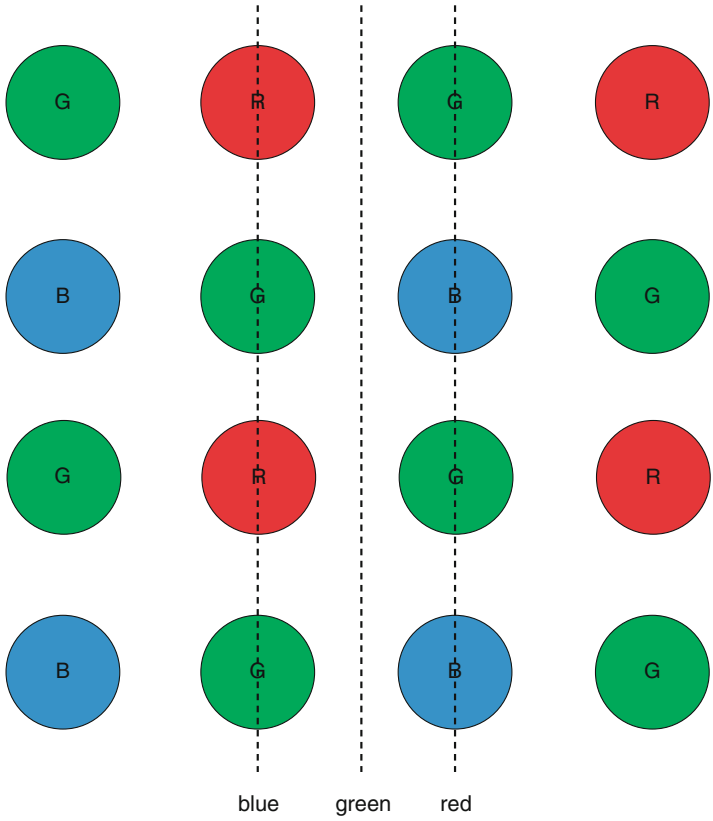


Fig. 3.35 Moire patterns from color filter array interpretation

lines: it calculates a horizontal gradient and vertical gradient as the difference between horizontally/vertically aligned pixels. If the algorithm finds a horizontal gradient, it uses vertically aligned pixels for interpolation; in the case of a vertical gradient, it uses the horizontally aligned pixels; and in the case of no gradient, it uses all local pixels.

Constant-hue interpolation assumes that hue within an object is constant. They are generally more expensive in computation time and memory and are therefore less likely to be used as part of the imaging chain. This approach uses the green channel to adjust the interpolated red and blue values for hue constancy. *Reconstruction-based algorithms* use assumptions about the correlation between channels or the image characteristics. Such approaches may minimize a cost function and apply Bayesian estimation or a Markov random field model.

3.6.2 White Balance

White balance is required to ensure that white elements of the image are not mis-rendered due to the color temperature of the illuminated light. The simplest approach to white balance is the *gray world assumption*—we assume that the average color of the image is gray, or

$$\bar{R} = \bar{G} = \bar{B} \tag{3.22}$$

To perform the white balancing, we find the average value of all the pixels in the image and then compute an adjustment coefficient for each color component:

$$G_R = \frac{1}{3\bar{R}}\left[\bar{R} + \bar{G} + \bar{B}\right], \tag{3.23}$$

$$G_G = \frac{1}{3\bar{G}}\left[\bar{R} + \bar{G} + \bar{B}\right], \tag{3.24}$$

$$G_B = \frac{1}{3\bar{B}}\left[\bar{R} + \bar{G} + \bar{B}\right]. \tag{3.25}$$

Figure 3.36 shows an example for which the gray world assumption fails to produce an accurate white balance. The tunnel is lined with an orangish brick; a row of larger gray bricks at the bottom of the image provides a natural gray reference to illustrate the magnitude of the overall white balance. Because the color of the small, orange bricks dominates, the algorithm assumes that this luminance distribution represents gray and shifts the larger rocks (and the rest of the image) away from true gray and toward orange.

An enhanced version of this approach directly applies the gray world model only if the average color falls within a specified region of the color space [Koi96]. Figure 3.37 shows the region for which the average scene color is assumed to be gray:

Fig. 3.36 An example of gray world white balance failure

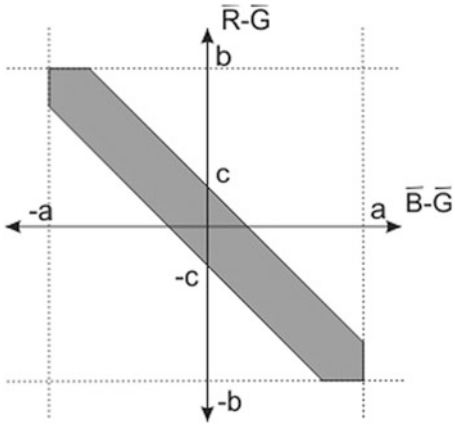


Fig. 3.37 A region of acceptability for gray world white balance

$$\begin{aligned} -a &< \bar{B} - \bar{G} < a, \\ -b &< \bar{R} - \bar{G} < b, \\ -c &< \left(\bar{R} - \bar{G}\right) + \left(\bar{B} - \bar{G}\right) < c. \end{aligned} \tag{3.26}$$

The region is defined relative to the color difference signals $\bar{R} - \bar{G}, \bar{B} - \bar{G}$. If the average color falls outside of this region, the R and/or B values in the image are adjusted so that the average falls within the acceptable region.

Kim et al. [Kim08] developed a method that fits the image into one of several standard illuminants. The CIE standard defines a number of standard illuminants and points in the color space that correspond to particular types of light sources.

They divide the image into blocks and discard blocks with low brightness because they do not contain much color information. They compare three types of color features: a modified gray world method, the white patch method, and color clustering. To apply the modified gray world method, they select blocks whose color is significantly different from surrounding blocks to reduce the chance of color casting. They identify white blocks by looking for blocks with very high brightness. They cluster the block colors to find a representative color. Given these three features, they identify the standard illuminant closest to the set of feature illuminants; they reject the match if it is larger than a threshold.

3.6.3 Sharpening

Most cameras apply a sharpening algorithm to non-raw images. People prefer the higher acutance provided by sharpening. The result may not be an entirely realistic rendering of the scene, but it is one that most people find pleasing.

We need to sharpen edges that appear in any orientation. We can do so using a form of the Laplacian operator for brightness:

$$\nabla^2 B = \frac{d^2 B}{dx^2} + \frac{d^2 B}{dy^2}. \tag{3.27}$$

This operation only identifies points at which the image brightness is varying rapidly. To sharpen an image, we want to add this result back into the original image. We also need to find a discrete form for the filtering operation. We can do so by generating each filtered pixel as a weighted combination of pixels in a region:

$$S(i,j) = \sum_{-1 \le i \le 1} \sum_{-1 \le j \le 1} c(i,j) I(i,j). \tag{3.28}$$

Our sharpening operator uses +9 at the center of the window and -1 elsewhere as shown in Fig. 3.38. The central +9 value compensates for the eight subtracted neighbor values and adds in the central pixel's value.

This sharpening filter has low overhead and is appropriate for implementation in the imaging chain. It can be performed in-place by writing the filter result back into the original image; the differences caused by overwriting the pixel values with sharpened values will be small. More sophisticated sharpening algorithms require

Fig. 3.38 A filtering window for sharpening

more memory as well as additional processing time and so are less suited to being performed on the fly. We will discuss more sophisticated sharpening algorithms in Section 4.resolution.

3.7 Image and Video Compression

Lossy compression is critical to the success of digital photography and video. This section considers compression algorithms for both still images and video.

3.7.1 Lossy Compression

Cameras generate lots of data; even with improvements in storage capacity, data consumes bandwidth and power. Image storage also makes use of lossless compression; we will consider file formats in more detail in Section 3.platform.io. But lossy compression provides much larger *compression ratios* to improve file size, bandwidth, and power.

Lossy compression in media relies on *perceptually aware coding*—our algorithms are designed to throw away parts of the data that are less likely to be observed by the human perceptual system. Perceptually aware coding is used in both audio standards like MP3 and visual standards such as JPEG and H.264.

Lossy compression for still and video images relies on somewhat different principles. Image compression reduces information by eliminating fine detail. This approach is well-suited to casual photography; it is not always the best approach for fine art. Video compression takes advantage of the fact that we cannot easily distinguish details in moving objects.

The design of a camera depends entirely on the choices of the designer. However, we need to *standardize* formats for images and video so that we can effectively use them: move them from camera to computer, run applications that read, and write the imagery. Standards committees are responsible for formulating standards for a range of technical subjects. Manufacturers are not required to meet these standards; however, they are generally required to satisfy certain compliance criteria in order to receive permission to use the trademarks associated with the standard. (They may also need to pay license fees for patents associated with the standard.)

3.7.2 Image Coding and JPEG

Figure 3.39 illustrates the key steps in the JPEG compression process [Wal91]; we will defer some details until later. The image is broken into 8×8 *blocks*; the discrete cosine transform is computed for each block; the block is quantized,

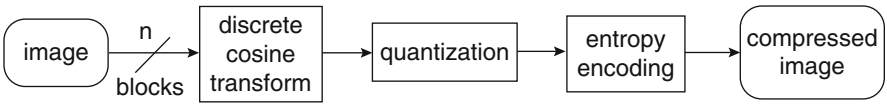


Fig. 3.39 The JPEG process

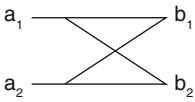


Fig. 3.40 Signal flow graph of the Cooley-Tukey butterfly

making use of the DCT data; entropy coding is then applied to reduce the size of the representation, resulting in a compressed image. JPEG decoding reverses the process: entropy decoding, dequantization, inverse DCT, and recomposition of blocks into the image. Let us consider these steps and the relationships between them.

JPEG relies on the *discrete cosine transform* (*DCT*) [Ahm74] to analyze the perceptual characteristics of the image suitable to lossy compression. The DCT is nearly optimal in several characteristics related to its encoding properties. DCT has also taken a life of its own as a primitive operation that is used in many other algorithms. We will discuss hardware and software implementations for the DCT in more detail in Sect. 3.8.6.

Given a sequence of values $x(i), 0 \leq i \leq N-1$, its discrete cosine transform $X(k)$ is

$$X(k) = \sum_{0 \leq i \leq N-1} x(i) \cos\left[\frac{\pi}{N}\left(i + \frac{1}{2}\right)\mathrm{k}\right], 0 \leq k \leq N-1. \tag{3.29}$$

The DCT can be written in several forms; this form is known as *Type II*. Note that the cosine term can be precomputed—it depends on i but not on $x(i)$.

The inverse transform—known as the *IDCT*—is

$$x(k) = \frac{1}{2}X(0) + \sum_{0 \leq i \leq N-1} X(i) \cos\left[\frac{\pi}{N} i\left(k + \frac{1}{2}\right)\right], 0 \leq k \leq N-1. \tag{3.30}$$

This form is known as *Type IV*.

Both the DCT and IDCT can be rewritten a recursive form known as the *butterfly*, illustrated in Fig. 3.40. This form was discovered by Cooley and Tukey and serves as the basis for the fast Fourier transform (FFT); the DCT is closely related to the discrete Fourier transform and therefore to the FFT. The butterfly computes two outputs from two inputs: $b_1 = a_1 + xa_2, b_2 = a_1 - xa_2$. The term x is known as the *twiddle factor*; in the case of the DCT, it corresponds to the cosine term; for FFT, it is $e^{-2\pi ik/n}$. In general, we multiply by coefficients, but they are not always shown in butterfly diagrams for simplicity. We can organize the 8×8 DCT into butterflies as shown in Fig. 3.41. The computation is performed in three stages, each with a smaller span of values.

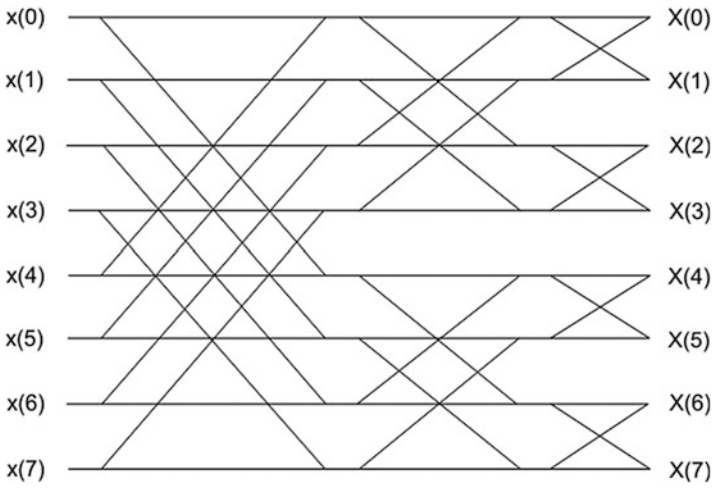


Fig. 3.41 The 8×8 DCT formulated as butterfly operations

We need a two-dimensional transform of the image. One of the useful properties of the DCT is that we can form the $N \times N$ 2-D DCT using two size N 1-D DCTs, one for the rows and the other for the columns:

$$X(k,l) = \sum_{0 \le i \le N-1} \sum_{0 \le j \le N-1} x(i,j) \cos\left[\frac{\pi}{N}\left(i+\frac{1}{2}\right)l\right] \cos\left[\frac{\pi}{N}\left(j+\frac{1}{2}\right)k\right], 0 \le k \le N-1. \tag{3.31}$$

The DCT does not by itself compress the image. It does, however, rewrite the contents of the block—known as *quantization*—to make it easier to identify content that can be removed for lossy compression. The DCT matrix is organized by spatial frequencies in both the horizontal and vertical dimensions: $X(0,0)$ corresponds to the DC value or the average value of the block; $X(N-1,0)$ represents the strength of the highest spatial frequency in the horizontal dimension; $X(0,N-1)$ represents the highest spatial frequency in the vertical dimension; and $X(N-1,N-1)$ gives the value of the highest spatial frequency component in both the horizontal and vertical dimensions.

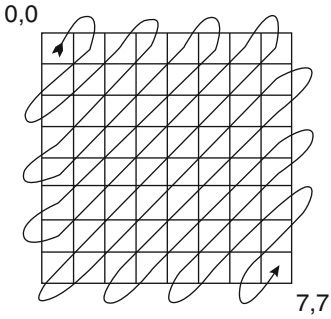
Fine detail corresponds to high spatial frequencies. If we want to reduce the fine detail in the image, we want to make the coefficients of the high spatial frequencies smaller. Quite a few different schemes can be used to quantize the DCT. The most general specification is as a matrix. Figure 3.42 shows a sample matrix from the JPEG standard designed for average image quality. The quantized DCT matrix is generated from a quantization matrix Q as

$$B(i,j) = \text{round}\left(\frac{X(i,j)}{Q(i,j)}\right). \tag{3.32}$$

Fig. 3.42 An example quantization matrix

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Fig. 3.43 The zigzag pattern for reading DCT coefficients



We want to design the quantization matrix to maximize its effect on the size of the encoded image while minimizing its effect on the image's quality. Reducing some DCT coefficients to 0 has particular advantages. A sequence of zeros can be very efficiently encoded using *run-length coding*—rather than writing out all the zeros in full two's complement representation, we can use a much more efficient code to represent the presence of n zeros.

We can maximize the impact of zeroing out certain coefficients in the *zigzag* pattern shown in Fig. 3.43. This pattern reads the coefficients in order of their spatial frequency. If we zero out coefficients at high spatial frequencies, the result will be to place zeros in the lower-right corner of the DCT matrix. The zigzag pattern will generate longer sequences of zeros that would, for example, read in row-major or column-major format.

The JPEG standard allows several different entropy coding algorithms to be used. The typical application applies run-length coding and then Huffman coding. The standard also allows arithmetic coding.

To summarize:

- DCT quantization directly influences the quality of the compressed image and indirectly influences the size of the compressed representation.
- Entropy coding directly influences the size of the compressed representation.

Most JPEG encoders provide a *quality* index to control coefficient quantization. Figure 3.44 shows an image encoded at several different quality levels; the original image is differenced against the image encoded at 2% quality.

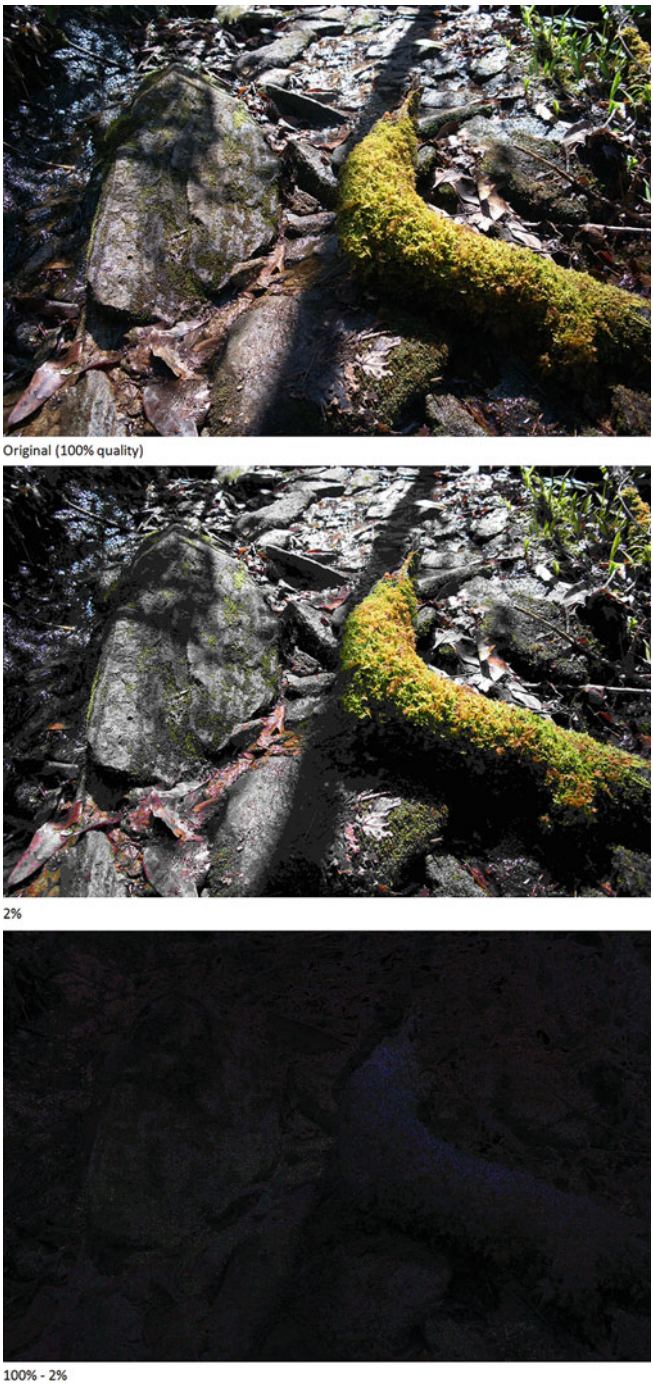


Fig. 3.44 Differences between images coded at different quality levels

The JPEG standard allows several other steps. The color space may be transformed to use YCrCb rather than RGB. If the color space is YCrCb, the chroma components (Cr and Cb) may be spatially subsampled to either half resolution horizontally or half resolution both horizontally and vertically.

The JPEG standard allows features to be used in various combinations. Today, the codification of a set of features is known as a *profile*. This practice was not fully formalized when JPEG was created. However, some common formats have been created. The most widely used format for the creation of JPEG files is the *JFIF standard* [Ham92]. The JFIF standard is compatible with the JPEG standard but specifies that files be written in a particular way. The aspect of JFIF most directly relevant to the image itself is the requirement to use YCrCb; JFIF also specifies the spatial relationship between the positions of pixels in the highest-resolution component and in the lower-resolution components.

3.7.3 Video Coding, H.264/AVC, and HEVC/H.265

Unlike image coding, in which JPEG is a dominant standard, several different video coding standards are in common use. At this writing, *H.264*, also known as *MPEG-4 AVC*, is used in a number of applications. *HEVC*, also known as *H.265*, is in the early stages of deployment.

In order to understand important features of modern video coders, we need to first outline some basic concepts in video coding.

At the heart of video coding are *block motion estimation* and *motion compensation*. As illustrated in Fig. 3.45, a frame broken into areas traditionally known as *macroblocks* and the motion of the objects in a subsequent frame is estimated. The macroblock is traditionally 16×16 although modern video compression standards allow for motion estimation on other sizes of blocks.

After transmitting the initial macroblock, we can transmit its movement in subsequent frames using a *motion vector* that is much smaller than the macroblock. We decode the image by applying the motion vector to the macroblock values and placing them in their new, compensated positions. Coding one frame in terms of another is known as *interframe coding*.

Motion estimation and compensation provide a great deal of compression but are not sufficient to fully encode the video stream: within a frame, motion estimation may fail to find a sufficient match; new objects may enter into the frame; and transitions such as cuts and dissolves change all the contents of the frame. We can form a complete coder as shown in Fig. 3.46 by computing the difference between the video stream and the decompressed form of the compressed stream. The result is a *residual signal* that is encoded using transform coding and quantization, much as in JPEG still image compression. The results of block motion estimation and residual signal compression are sent to the entropy coder to reduce the size of their representation.

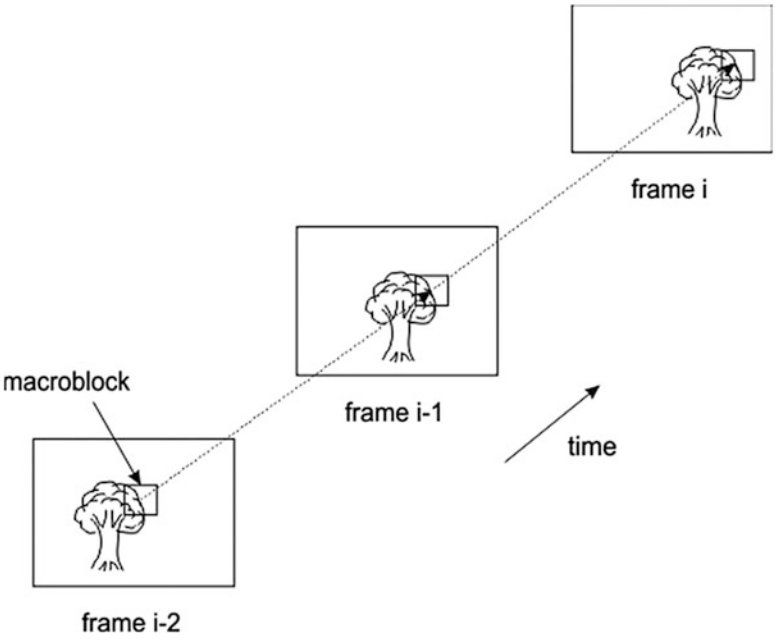


Fig. 3.45 Block motion estimation

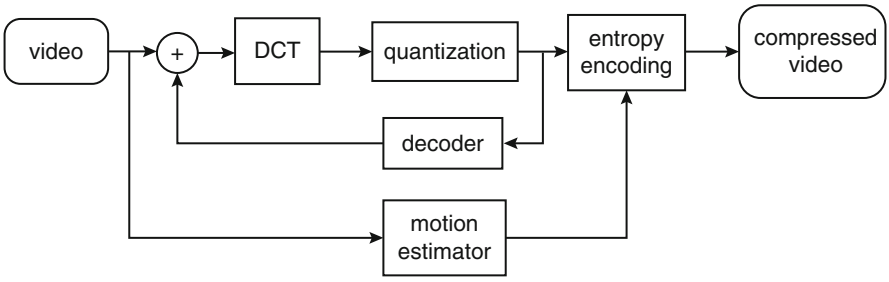


Fig. 3.46 Organization of a typical video encoder

Figure 3.47 shows the block diagram of a typical decoder. After entropy decoding, the information is separated into motion vectors and DCT coefficients. Reconstructed frames are saved in a *frame store* so they can be used by other parts of the compression system.

The compressed video stream relies on at least one frame that is not encoded in terms of other frames. Thanks to history, we refer to such frames as *I frames* for *intraframe*. A frame whose motion is predicted by past frames is known as a *P frame* for *predictive*. We can also analyze motion using frames both before and after the frame under consideration—we use buffers to hold the sequence of frames and

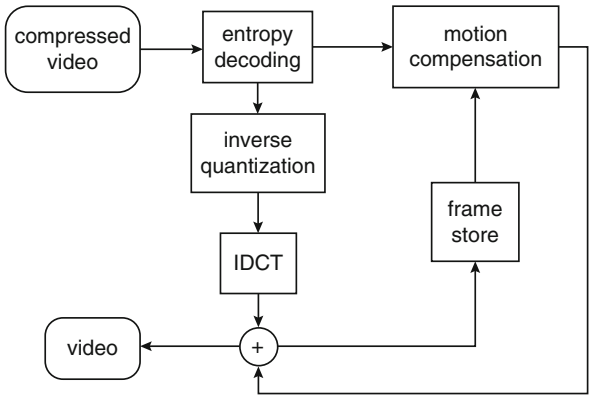


Fig. 3.47 Organization of a typical video decoder

wait to generate the compressed output until we have all the required frames. Such frames are known as *B frames* for *bidirectional*.

Block motion estimation (often referred to as *BME*) assumes translational motion. BME is powerful because it is both sufficiently accurate and easy to compute [Net79]. Given two macroblocks M_1, M_2, we can find the motion vector from M_1 to M_2 by computing the sum-of-absolute differences of the pixels between the macroblocks:

$$SAD_{12} = \sum_{0 \le j \le B} \sum_{0 \le i \le B} |M_1(i,j) - M_2(i,j)|. \tag{3.33}$$

We choose M_2 to have some offset $\langle x, y \rangle$ relative to M_1 in the image frame. We find the motion vector for several different offsets and select the offset with the lowest SAD value to determine the motion vector.

A full-search computation of the motion vector is very expensive. Each SAD requires B^2 difference/absolute value/sum operations; if $B = 16$, then each SAD requires 256 operations. The total number of operations for full search depends on the distance of the search D operation from the original location. Full search of macroblocks requires D^2B^2. Given that search regions of radius 16–64 may be necessary, full search is too expensive for most applications—it takes too much time, requires too many memory accesses, and consumes too much power.

A number of heuristic search algorithms have been developed to reduce the cost of motion estimation. Dozens of such search algorithms have been proposed; three popular alternatives are *three-step search*, *four-step search*, and *diamond search*.

Three-step search has been proposed in several variations. One enhanced verison [Li94] is illustrated in Fig. 3.48. The first step searches eight exterior points, the middle point, and eight points around the middle. If the minimum cost is at the center, search stops. If the minimum-cost point is one of the neighbors of the center point, an additional search step of the eight neighbors around that minimum is performed. Otherwise, a set of eight points in a tighter radius around the minimum are searched, followed by a third round of eight points at adjacent pixels.

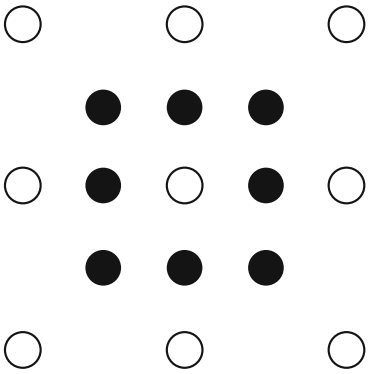


Fig. 3.48 Three-step search for motion estimation

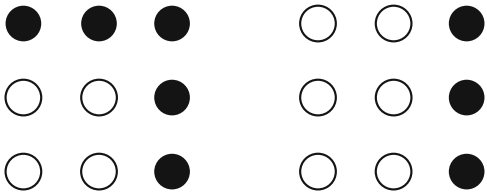


Fig. 3.49 Four-step search for motion estimation

Four-step search [Po96] starts, as with three-step, by checking nine points in a 5×5 window. The window remains at the same size for the second step, but the search area is modified as shown in Fig. 3.49, depending on whether the minimum-cost point is along a side or at a corner. The third step uses the same strategy. The fourth step searches a nine-point pattern of adjacent pixels in a 3×3 window.

Diamond search [Tha98] starts with nine search points arranged in a diamond pattern as shown in Fig. 3.50. As with four-step search, the second step adds points in a pattern depending on whether the minimum-cost point was on a face or a vertex. The third and final step searches the four internal points of the previous diamond.

Some motion estimators perform *subpixel motion estimation* by interpolating pixel values. We can estimate intermediate pixel values using standard techniques: $I(i+0.5,j) = \frac{1}{2}[I(i,j) + I(j+1)]$, etc. We can then add these estimated pixels to the motion estimation problem to give more accurate motion vectors.

Video encoders define a format for the output of the encoder; the same format is used by the decoder. Since multimedia requires both audio and video, the complete representation includes a video layer, an audio layer, and a system layer that records the synchronization between them.

Video coders can operate in either *variable bit rate* or *constant bit rate* mode. A basic video coder will generate a variable number of bits at its output as the video content varies: some frames may require more bits than others; some parts of a frame may require more bits than others. However, highly variable bit rates make both storage and network transmission more difficult. We can adapt a coder to constant bit rate mode by introducing a feedback loop from the entropy coder,

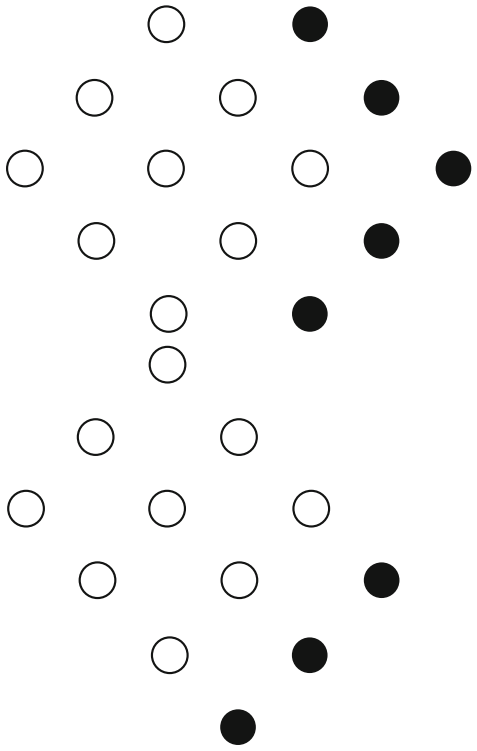


Fig. 3.50 Diamond search for motion estimation.

which knows the number of bits being generated, and the rest of the coder. However, naïve constant bit rate coding introduces variances in the quality of the generated video. In the simplest case, if the encoder is allocated a fixed number of bits per frame, it could encode the top-left region of the frame at the highest quality and successively reduce image quality as it moves toward the bottom right and runs out of bits.

Sulllivan and Baker [Sul91] proposed the use of the *Lagrange multiplier* method to optimize the rate-distortion characteristics of block motion estimation. In particular, they were interested in variable-sized block motion estimation, in which some areas would be encoded using larger blocks that give less accurate information about the motion within that region. They observed that once the distortion of macroblocks has been estimated, the distortion of larger areas can be easily computed from the macroblock values; the distortion estimation process can be modeled as a tree. They formulated the optimization of rate $R(B)$ and distortion $D(B)$ of a bit allocation B as an unconstrained problem using a Lagrange multiplier λ:

$$\min_{B \epsilon S} [D(B) + \lambda R(B)] \tag{3.34}$$

for the set of possible bit allocations S. Larger values of λ result in lower rates, while smaller values reduce distortion. The global rate distortion can be minimized by minimizing the Lagrangian for each block b_k:

$$W_k(b_k) + \lambda b_k \tag{3.35}$$

for the block distortion W_k.

H.264/MPEG-4 AVC, commonly known simply as H.264, is a widely used video compression standard [Wie03]: it is used for Blu-Ray™ discs, in many surveillance cameras, and in many consumer video cameras. The complicated name comes from history. Several generations of video coding standards had been designed, with one lineage for consumer video and another for teleconferencing. H.264/MPEG-4 AVC was created to unify these different applications. The newer HEVC standard provides improved compression ratios using several techniques including the sizes of several types of regions and improved prediction within and between frames.

We will concentrate here on the video layer of H.264 known as the *video coding layer* (*VCL*). H.264 also uses a *network abstraction layer* (*NAL*) built on top of the VCL to allow the data to be used in a variety of applications, including videoconferencing, broadcast/recording, and streaming.

Figure 3.51 gives a block diagram for the H.264 encoding process. The middle part of the block diagram operates as a decoder—after reconstructing the decoded

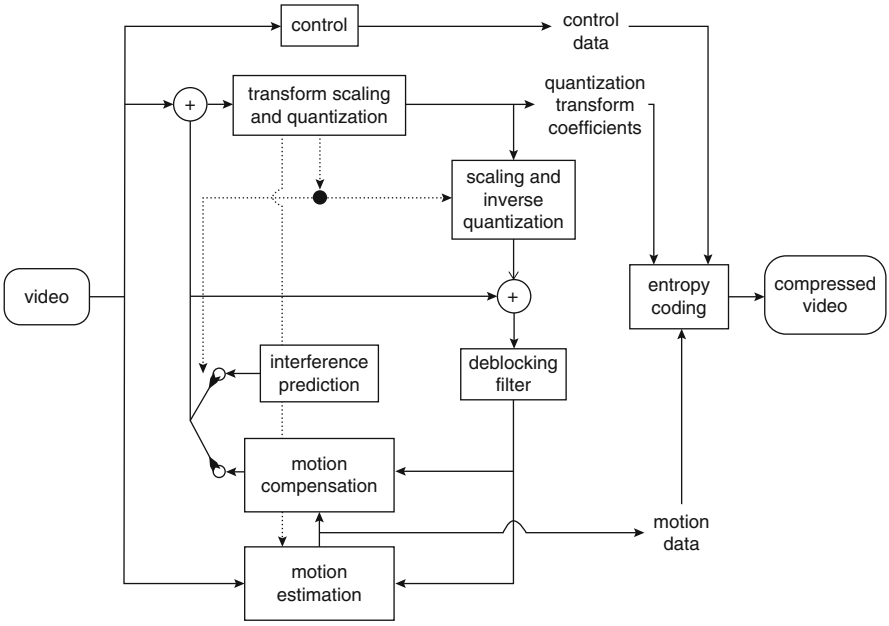


Fig. 3.51 Block diagram of H.264 encoding

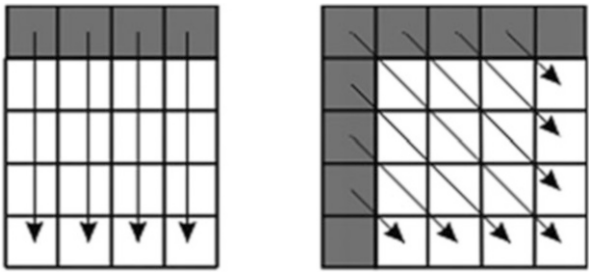


Fig. 3.52 Intra-prediction

image, the result is compared to the original image and the difference encoded to improve the quality of the result.

H.264 provides flexible mechanisms for block motion estimation. It can perform motion estimation over blocks of several sizes: $4 \times 4, 4 \times 8, 8 \times 4, 8 \times 8, 8 \times 16, 16 \times 8, 16 \times 16$. The ability to use different-sized blocks for motion estimation allows trade-offs between estimation quality and bit rate; H.264 uses Lagrangian methods to optimize rate distortion [Wie03B]. It can also store several reference pictures for motion estimation and compensation; this feature was motivated by periodic motion, in which a sequence of blocks may appear repeatedly in the sequence.

H.264 uses the YCrCb color space and 4:2:0 sampling (half the sampling rate both horizontally and vertically) with eight bits per sample.

Intra-prediction encodes information on a block using pixels from neighboring blocks in the same frame—pixels in the block are filled with copies of neighboring pixels. A 4×4 predictor is used for luminance blocks and supports nine modes, each of which fills the predicted block from different surrounding directions. Figure 3.52 shows two of the nine prediction modes: pixels are copied vertically from the row above the block; pixels are copied diagonally down and right. A separate mode using larger blocks is designed to efficiently encode large, uniform regions. This mode supports four different directions and can be used for 16×16 luminance blocks or 8×8 chrominance blocks. Prediction is performed at ¼ luma sampling. The intra-prediction can also be bypassed with a mode that directly sends the samples.

Transforming coding of the residual does not use the DCT [Mal03]. It instead operates on a 4×4 block and uses a transform matrix with all integer values. The small block size is used because the residual signal has less spatial correlation than does a standard image. The integer transform coefficients allow the operation to be performed efficiently.

A *deblocking filter* smooths out block boundaries to minimize the visual effects of mismatches between the visual characteristics of adjacent blocks. H.264 requires the use of a deblocking filter to avoid using blocky frames in the compensation loop. The deblocking filter first filters vertical macroblock edges and then horizontal edges. The filter takes as input eight pixels, four from each side of the edge; it updates six pixels in a luminance block or four in a chrominance block. Boundary strengths are used for adaptive filtering; the boundary strength of a chroma block is determined by the strength of the corresponding luminance boundary. The filter examines pixel values

around the block boundary to determine if smoothing should be applied; it looks for variations across the boundary that are significant enough to need smoothing but not so large that they probably represent an object boundary in the image.

3.7.4 Quality Assessment of Compressed Images

Peak signal-to-noise ratio (*PSNR*) is often used to evaluate image and video algorithms. However, PSNR does not reflect any perceptual characteristics of the visual system. PSNR is typically defined by *mean-squared error* from the original image I to its noisy or compressed version J:

$$\text{MSE} = \frac{1}{N^2}\sum_{0\le i<N}\sum_{0\le j<N}(I(i,j)-J(i,j))^2. \tag{3.36}$$

The PSNR is, in turn,

$$PSNR = 10\log\frac{\text{MAX}^2}{\text{MSE}} \tag{3.37}$$

where MAX is the maximum value of the original image. This simple formula is easy to compute but does not weight image characteristics in a way that takes into account perception. We discussed the assessment of image quality in Section 2. quality but that discussion assumed ideal images. Since compression algorithms discard image information, we need additional tools to understand how lossy compression affects image quality.

The *structural similarity index model* (*SSIM*) [Wan04, Bov13] is a widely used metric that takes into account perceptual criteria but is also easy to compute. SSIM compares two images—an ideal reference image and the image to be tested. It compares $N \times N$ windows from each image; we typically compare several windows from each image. If the two image patches are x and y, each with an average μ, variance σ, and covariance σ_{xy}, then their SSIM is given by

$$\text{SSIM}(x,y) = \left[\frac{2\mu_x\mu_y + c_1}{\mu_x^2+\mu_y^2+c_1}\right]^{\alpha}\left[\frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2+\sigma_y^2+c_2}\right]^{\beta}\left[\frac{\sigma_{xy}+c_2/2}{\sigma_x\sigma_y+c_2/2}\right]^{\gamma}. \tag{3.38}$$

In this formula, α,β,γ are weighting factors for the component. The additional terms are $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ where $L = 2^{\text{bits per pixel}}$ and the typical values for the coefficients are $k_1 = 0.01, k_3 = 0.03$.

3.8 Computing Platforms

Digital cameras are complex computer systems as well as optical systems. We refer to the computer that underlies a complex system as its *computing platform* or simply *platform*. As we have seen, cameras must perform complex computations in the imaging chain. Those computations also vary widely in their characteristics—digital filtering, for example, is very different from file system operations. Cameras must perform under real-time constraints—*deadlines*—in order to avoid dropping data and generating bad images. Most cameras also operate under power consumption limitations.

In this section, we will consider the computing platforms for digital cameras. A thorough discussion of digital camera computing platforms would occupy several books. However, a basic understanding of camera platforms helps us to understand some of the design decisions and trade-offs in camera design. We will also consider the software required to operate a digital camera.

3.8.1 Cameras as Heterogeneous Multiprocessors

The result of the stringent requirements on cameras—performance, power, weight, and cost—is that most camera platforms are *heterogeneous multiprocessors,* collections of several different types of processors interconnected together. Many embedded processors are heterogeneous [Wol08] because heterogeneity is the most effective way to simultaneously meet real-time performance, power, and cost constraints. The *processing elements* in a heterogeneous multiprocessor are either programmable or fixed-function units known as *accelerators*. We will look at the design of two important image and video accelerators in Section 3.platform. accelerators. In addition to image processing, digital cameras need to perform a variety of functions that are common to interactive computer systems, notably file system management and user interface. The *host processor*, typically a RISC processor, is responsible for such tasks.

While heterogeneous multiprocessors confer many advantages, ease of programming is not one of them. The different types of processing elements often use different instruction sets. The communication between these processors often requires specialized mechanisms. Given the large volumes of software in modern cameras, programming these cameras has become a complex task in itself, even once the algorithms they perform are well-understood.

Several camera manufacturers have designed their own image processors; these processors are generally proprietary and relatively few details on them are available. We will introduce two different chips used for digital multimedia systems as examples of the wide range of possibilities in the digital camera platform design space.

The Texas Instruments AM572x Sitara processors [Tex16] include a dual ARM Cortex-A15 microprocessors, two dual ARM Cortex-M4 processors, two C66X digital signal processors, an image and video accelerator subsystem (IVA-HD), a 3D GPU, and a 2D graphics accelerator. It also includes on-chip memory. The Cortex-A15 units are organized as an MPCore multiprocessor [ARM09], which provides a snooping mechanism for cache coherency; each processor includes a Neon SIMD coprocessor and floating point. The C66x is a very long instruction word (VLIW) processor which can be used for audio, imaging, and video processing. The Cortex-M4 CPUs provide hardware division and single-cycle multiplication. The IVA-HD system includes a set of accelerators for video encoding and decoding.

Smartphone processors combine many different architectural forms to provide high performance at low power levels: RISC clusters, digital signal processors, vector units, VLIW units, GPUs, and accelerators.

GPUs are increasingly common in digital camera platforms, particularly those hosted on smartphones. GPUs provide enormous numerical processing power in a relatively compact area. The graphics problems they were originally designed to solve are in some sense the inverse problem of imaging. However, in both cases the image can be broken up into relatively independent groups of pixels, allowing computations to be performed with an embarrassing level of parallelism. The NVIDIA Jetson TX1 [NVI14B] includes the Maxwell GPU. Maxwell [Nvi14] includes 640 cores organized into four streaming multiprocessors. Cores can perform floating-point arithmetic [Whi11]; the large number of floating-point units gives GPUs their tremendous numerical computational power. Programs to run on the cores are specified as threads; during execution, the GPU groups threads into *warps* for scheduling purposes. Each streaming multiprocessor includes a 16,384 X 32-bit register file; each thread can access up to 255 registers at a time. The set of streaming multiprocessors also share a separate 96 KB memory. The TX1 also includes a cluster of ARM Cortex A57 organized as an MPCore cluster. The Cortex A57 provides a 64-bit architecture and a floating-point unit.

Some GPUs provide limited-precision floating point, such as 16-bit floating-point arithmetic. These smaller formats provide two benefits: operations consume less power and smaller values result in more available registers and local memory. Their limited dynamic range does affect the accuracy of results in some algorithms; however, these smaller floating-point formats may be useful in some applications. We will discuss an example in Sect. 4.11.1.

3.8.2 Buffering

The design of *buffering* in the platform is critical to the satisfaction of design requirements. Inadequate buffering can reduce the image throughput; for still cameras, this means less frequent capture of fast action, while for video, this means reduced frame rates. Solving throughput problems by adding too much

Fig. 3.53 Buffers in the image chain

memory results in excessive power consumption. Buffers are required at several points in the imaging chain; these buffers vary in their purpose and organization.

Figure 3.53 shows the placement of buffers in the image processing chain. Let us consider these buffers one at a time.

The buffer between the image sensor and the image operations (Bayer pattern filtering, sharpening, etc.) is relatively simple. Image operations must operate on several adjacent lines in the image, but they do not require the entire image. The rate at which data is consumed by the image operations is constant, so we can easily determine the amount of memory required. As a result, this buffer is typically much smaller than an image frame. Because this buffer is small, it may be built using dedicated static RAM rather than bulk dynamic RAM (DRAM).

Compression and storage operations are more variable in both execution time and data volume. Execution time for some algorithms may vary; compression may also result in differing volumes of compressed data, which results in varying amounts of time required for transfer. The amount of data in the buffer at any given time t depends on the history of input to the buffer and output from the buffer:

$$B(t) = \sum_{o \le i \le t} [\mathrm{Out}(i) - \mathrm{In}(i)]. \tag{3.39}$$

The buffer between the image operations and compression is particularly important in still cameras for *bursts*. Action photographers often take a sequence of photos and select their preferred image later; we will also see that some image enhancement algorithms make use of image sequence bursts to minimize motion between images. If the compression system is not fast enough to keep up with the frame rate generated by the image sensor, then the size of the buffer at the compression unit's input determines the maximum length of a burst—the burst must pause when the buffer becomes full.

Similarly, if the storage system is not fast enough to keep up with the compression unit's output, the size of the compression/storage buffer will limit the burst sequence length. For video capture, the storage system must be fast enough to keep up with compression—limiting the length of a video clip would be acceptable only in, perhaps, an ultrahigh frame-rate camera. However, the data volume produced by the compression unit may vary. Even if the storage system can keep up on average, buffering is required to avoid data loss. This type of buffer is known as an *elastic buffer*.

The buffers at the input and output of the compression unit are typically built from bulk DRAM. These buffers are considerably larger than the one used for image operations; DRAM provides both lower cost and lower power consumption than static RAM. This bulk DRAM may also be shared by the processors for

program and data. If DRAM is used by multiple processing elements, *contention* between the units must be factored into performance calculations to ensure that units are not starved.

3.8.3 Input and Output

Most modern cameras use electronic displays. Some professional and enthusiast cameras use optical viewfinders, either rangefinders or single-lens reflex finders. Optical systems operate at the speed of light but do not take advantage of the image sensor to pre-analyze the image.

The key design requirement for electronic displays is low latency—the delay from image sensor to the display screen must be very small so that the photographer can assess composition with moving objects. Low display latency requires careful design to minimize buffering. Generally speaking, the resolution of the display is smaller than that of the image sensor. However, the ratio of display to image sensor resolution can vary widely: smartphone screens are a closer match to the sensor resolution than are many mirrorless enthusiast cameras. Subsampling the image sensor lines both reduces read time and matches the vertical resolution of the viewfinder image to the screen. Digital filters can be used to reduce the image's horizontal resolution. Buffering latency can be minimized by matching the display frame rate and the image sensor capture rate.

At the other end of the I/O chain, we have mass storage. Both compact flash (CF) and secure digital (SD) cards are used in camera, but SD is more widely used in modern cameras. Compact flash is based on the electrical standard of IDE magnetic disk drives, although its pinout is different. CF originally offered larger storage and higher bandwidth than SD, but modern SD variants are vastly improved on both fronts. At this writing, versions of SD support up to 2 TB of storage and transfer rates of 312 MB/sec.

The characteristics of flash devices lead to some interesting characteristics. The write time of flash memory is considerably longer than read times. The transistors used to store data wear out with multiple writes. Combinations of software and firmware perform error correction as well as identify bad bits and swap in spares. However, the read time of flash memory can increase with use—the error correction algorithms used require more execution time as the number of bad bits increases.

3.8.4 File Formats

Cameras must create image data in file formats that can be used by general-purpose computers as well as many other devices. Many printers, for example, accept flash cards and can print directly from the card's contents. A number of image-related file

formats have been developed; some date back to the 1980s. We will survey a few widely used formats.

The TIFF format [Ald88] predates the JPEG standard. While the standard allows images to be stored in several styles, both compressed and uncompressed, it is typically used today to store images without lossy compression. The pixels can be stored directly or losslessly compressed using the LZW algorithm. TIFF allows the number of bits per sample/pixel and a color map to be specified. Since it predates modern color management standards, it does not directly conform to them. TIFF also provides for *tags* to record metadata. A tag is a $<$ *name, value* $>$ pair; TIFF defines some tag names and allows users to create their own tag names. Example TIFF predefined tags include image width and height, image orientation, image data location, image title, etc.

The JFIF file format we described in Section 3.JPEG is not often used on its own. The Exif format [JEI02] is commonly used to contain the JFIF data; many files with the *.jpg* format are, in fact, in the Exif format. Exif also includes an audio file standard; we will concentrate on the image format. Exif allows one file to contain several versions of an image as well as tags. The *primary image* can be saved in TIFF or JPEG formats. The file can also contain a *thumbnail* image—a small version of the image. Thumbnails are often used for quick display of the image; not only does the thumbnail have fewer pixels but it is often stored in a format that is simpler to decode than is JPEG. Exif allows both compressed and uncompressed thumbnails although a compressed thumbnail cannot be combined with an uncompressed primary image; an Exif-specific format is used for compressed thumbnails. Exif defines a variety of tags and allows user-defined tags. Examples include GPS information, x and y resolution, date and time, make/model/software of the image recording device, etc.

The DCF standard [Cip10] builds upon the Exif standard; it concentrates on the relationship of data to the storage media, writers, and readers. The creation of DCF was motivated, in part, by the ad hoc development of MP3 storage methods. The MP3 audio player was not originally considered in the MPEG-1 standard that defined MP3. MP3 audio was originally collected on computer hard disks and CDs. Because no standards existed for the use of MP3 files, even simple playback devices were required to read the entire directory structure of the storage medium in order to determine where playable files may be located. DCF specifies that image data be kept in the DCIM directory, which is to be stored within the root directory. File names are to be eight characters long not counting the extension; the first four characters are uppercase alphabetic; the last four characters are numbers in the range "0001"–"9999." DCF specifies several characteristics of writers, for example, that at most 900 DCF directories may be created under the image root directory. The standard also specifies required characteristics of DCF readers, most importantly that they be able to detect the directories on a DCF-compliant medium and display the files in a given specification.

Camera manufacturers often define *raw image formats* for their cameras. These raw formats are generally proprietary; many of the raw formats have been reverse engineered. Although the name implies that the data in the file is the original pixel

values without modification, some raw formats in fact perform lossy data compression and do not preserve all the raw image data.

3.8.5 Operating Systems and File Systems

The dominant smartphone operating systems, iOS and Android, support digital camera APIs. The μITRON operating system [Tak02] is widely used in digital cameras and other consumer electronics devices. It provides priority-based scheduling of tasks.

The FAT32 file system [Mic00] is widely used for removable flash cards in digital cameras and other consumer electronics devices. FAT32 was developed by Microsoft as an extension of its earlier FAT and FAT16 file system. It supports drives of up to 2 TB. FAT32 can be implemented in a relatively small amount of code and provides a robust file system interface. *Formatting* a file system establishes the basic file system data structures, such as a root directory, on the storage medium.

Flash memory wears out with writing [Wol17]. Many devices use a *flash translation layer* to optimize the use of the flash memory. Directories are most liable to fast wearing since they are modified much more often than are the sectors of typical files—a file's directory must be modified any time the file itself is modified. Flash translation layers perform *wear leveling* by occasionally moving directories to different locations in the flash drive. One consequence of the wear properties of flash memory is deleting all files by formatting rather than deleting individual files which is highly preferable—formatting requires only a single set of writes while deleting individual files results in a large number of file writes.

3.8.6 Accelerators

A great deal of image computation is performed by dedicated hardware. The primary reason for this architectural choice is power/energy consumption—hardwired units generally consume less energy per operation and less total power than an equivalent programmable unit. Given that most cameras operate on batteries, power consumption is a key concern.

Many digital cameras include hardwired units for both JPEG and video compression. The design of complete implementations of these standards is beyond our scope. We will concentrate here on the design of two types of units useful in image and video compression: DCT and block motion estimation.

We introduced the butterfly operation for DCT in Sect. 3.6.2. A hardware butterfly unit includes an adder, a negation unit, a multiplier, and a twiddle factor

coefficient. We can perform all the DCT operations in fixed-point arithmetic: both the input and output have limited dynamic range; the intermediate operations do not include divisions, so no hidden dynamic range excursions occur; the twiddle factors are in the range [0, 1] so we can perform operations with an implicit decimal point at the head of the twiddle factor. We can use the butterfly structure to perform the inverse DCT, but higher accuracy computations may be required.

We should note that the difference in hardware characteristics—area, performance, and power consumption—between an adder and a multiplier is not as great as many people think. While multipliers are inherently more complex than adders, modern VLSI processors make single-cycle multipliers feasible for a wide range of bit sizes; multipliers for the word sizes used in digital camera image processing are very reasonable.

A number of DCT algorithms designed for hardware efficiency—particularly in the 8-point DCT—have been proposed. We will describe a few here to illustrate the range of possible solutions. We can estimate the hardware cost of a DCT algorithm using the number of multipliers and adders it uses; however, wiring complexity and other factors also contribute to overall hardware cost.

Loeffler et al. [Loe89] developed an algorithm for an 8-point DCT that requires 11 multiplications and 29 additions. The algorithm is shown in Fig. 3.54. The order of the inputs and outputs are shown at the left-hand and right-hand ends of the structure, respectively; note that the outputs appear in a different order. The algorithm makes use of three kinds of units:

- A standard butterfly performs $O_0 = I_0 + I_1, O_0 = I_0 + I_1$.
- The c box performs $O_0 = I_0 k \cos\frac{n\pi}{2N} + I_1 k \sin\frac{n\pi}{2N}, O_1 = -I_0 k \sin\frac{n\pi}{2N} + I_1 k \cos\frac{n\pi}{2N}$. This box can be rewritten with a common factor to reduce its effort to three multipliers and three additions. Given the form $y_0 = ax_0 + bx_1, y_1 = -bx_0 + ax_1$, the formulas can be rewritten as $y_2 = a(x_0 + x_1), y_0 = (b - a)x_1 + ay_2, y_2 = -(a + b)x_1 + ay_2$.
- The open box performs $O = I\sqrt{2}$.

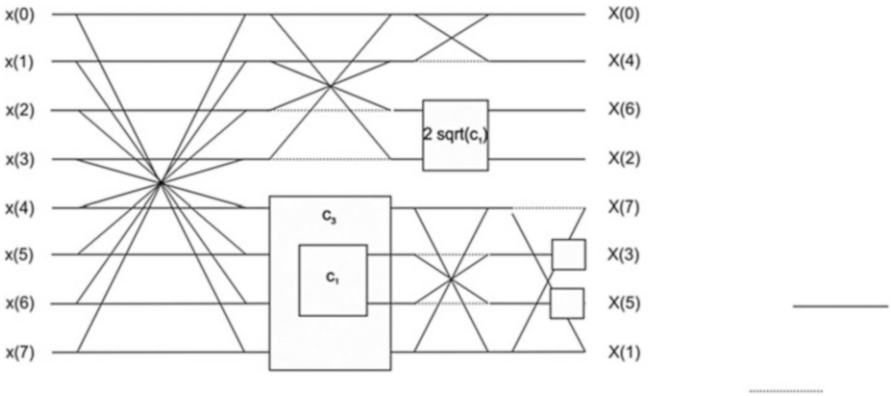


Fig. 3.54 An algorithm for DCT

Kok [Kok97] developed a recursive algorithm for even-length DCTs. This algorithm requires 12 multipliers and 29 adders for an 8-point DCT; it is also designed to have a very regular structure. The even output $C(i)$ is given by

$$C(i) = \sum_{0 \le n \le \frac{N}{2}-1} p(n) \cos\left[\frac{2\pi}{4n}(2n+1)2i\right], \tag{3.40}$$

$$p(n) = x(n) + x(N-1-n)\; n\,\epsilon\left[0, \frac{N}{2} - 1\right]. \tag{3.41}$$

The odd output $D(i)$ is defined in terms of $D^{'}(i)$:

$$D'(i) = \sum_{0 \le n \le \frac{N}{2}-1} q(n) \cos\left[\frac{2\pi}{4n}(2n+1)2i\right], \tag{3.42}$$

$$q(n) = x(n) - x(N-1-n)2\cos\left[\frac{2\pi}{4n}(2n+1)\right] n\,\epsilon\left[0, \frac{N}{2} - 1\right], \tag{3.43}$$

$$D'(0) = 2D(0). \tag{3.44}$$

Guo et al. [Guo92] developed an algorithm that substituted table lookup from ROM and adders for multipliers.

H.264 does not use the DCT but instead uses a transform with integer transform coefficients [Mal03]. The DCT's coefficients are irrational so that a DCT followed by its inverse may not result in exactly the same values returned. Using integer coefficients eliminates this problem.

The simplest way to implement a 2-D DCT is as a 1-D DCT, a transposition buffer, followed by a second 1-D DCT. This operation is particularly useful if hardware costs allow for only one DCT unit. However, direct 2-D DCT algorithms can provide more efficient implementations.

Cho and Lee [Cho91] developed a 2-D DCT algorithm based on N 1-D DCT modules as well as butterfly adders and shifters. They then reformulated the signal flow graph so that only half of the DCT modules were operational at any given time. As a result, they could use multiplexers to reformulate the algorithm to require $N/2$ 1-D DCT modules. Their algorithm operates on real numbers.

Lee et al. [Lee97] made use of complex arithmetic to develop a regular 2-D DCT algorithm. Their algorithm operates in three phases: pre-addition consisting of butterfly computations and multiplication by $1/\sqrt{2}$, complex DCT and rotation, and a butterfly postaddition. They also showed how to fold their architecture from a fully parallel form requiring four complex DCT units to a pipelineable unit that used transpose memories, multiplexers, and circular shifters in addition to the complex DCT and butterfly units. They showed that their IDCT architecture gave lower mean-square error values than did previous approaches. They showed that their folded architecture required four 1-D DCTs, four 4×4 trams[pse,e,proes. 76 adders, and four constant multipliers, totaling 402,048 transistors.

The key challenge in the design of motion estimation engines is memory bandwidth. The sum-of-absolute-difference operator is relatively simple, but a

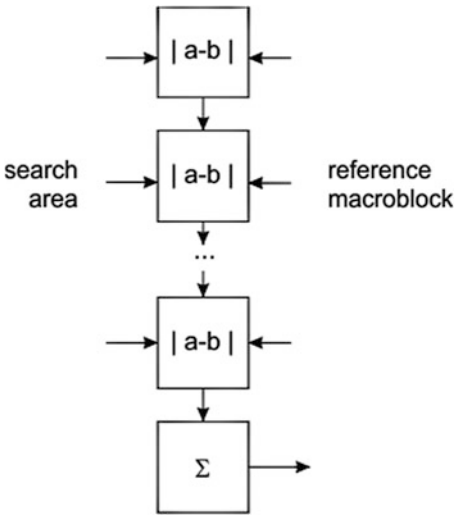


Fig. 3.55 A systolic motion estimation architecture

huge number of operations are performed. Efficient motion estimation engines use local registers to store values and carefully schedule operations to minimize the number of times that a given pixel must be fetched.

Komareck and Pirsch [Kom89] developed systolic architectures for motion estimation. These architectures use local communication links between arrays of processing elements. For example, Fig. 3.55 shows a one-dimensional systolic array. The reference macroblock and search area pixels are pumped into the array as wavefronts, each staggered from top to bottom. The $|a-b|$ processing elements pass their results to the summation block; a separate block is used to select the minimum-error motion vector. This unit requires $N(2p+1)(2p+N)$ clock cycles where N is the size of the macroblock in each dimension and p is the radius of the search area.

Yang et al. [Yan89] developed a motion estimation architecture designed for full search. The architecture is illustrated in Fig. 3.56; it includes 16 sum-of-absolute-difference (SAD) processing elements whose outputs feed a comparator. Their architecture schedules operations on pixels from the two macroblocks: a represents the reference macroblock while b represents the search area. Successive $a(i,j)$ values are broadcast to all of the absolute-difference operators, while each $b(k,l)$ value is shifted from one unit to the next. On the 15th cycle, for example, the absolute-difference units are processing $|a(0,15)-b(0,15)|, |a(0,14)-b(0,15)|, \cdots, |a(0,0)-b(0,15)|$. Each SAD unit computes the error for a different candidate motion vector. To accommodate the larger access region for b, the architecture includes two inputs for different parts of the b range and multiplexers to select the appropriate value. For a search area of $[-7,8]$ horizontally and $[-8,8]$ vertically, this architecture requires 4367 cycles [Dut96].

Dutta and Wolf [Dut96] extended this architecture to allow programming for algorithms other than full search. They added an interconnection network to

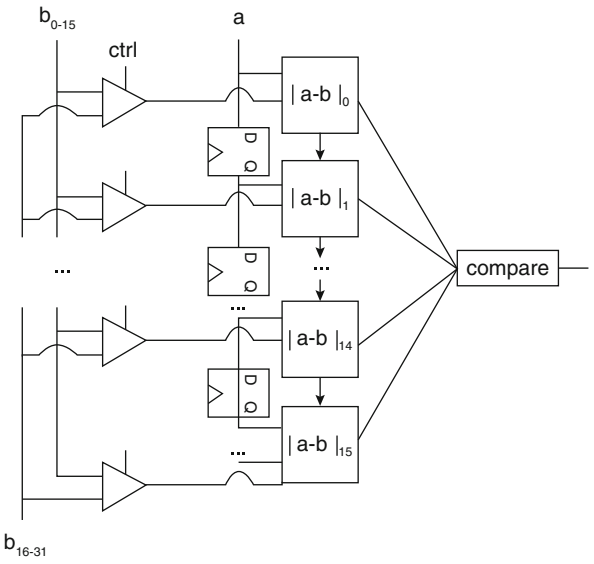


Fig. 3.56 A block motion estimation architecture

connect the memory elements to the processing elements, a multi-ported memory, and a programmable controller. When built with a generalized-cube network, a three-step search of distance ratio 4 : 3 : 1 required 2600 cycles.

Yang et al. [Yan05] analyzed the power consumption and performance of hardware and software implementations of motion estimation. They compared eight algorithms: full search, one-dimensional full search, three-step, four-step, diamond, modified log [Kap85], alternating pixel decimation, and subsampled motion field with APDS [Liu93]. They found four algorithms to provide the lowest power consumption in both hardware and software realizations: modified log, three-step, four-step, and subsampled motion field with APDS.

3.9 Image Characteristics and Image Capture

Not all scenes provide high-dynamic range; however, some scenes exhibit very large contrasts between light and dark. Mixed indoor-outdoor lighting, common in surveillance, often provides very wide dynamic ranges between indoor areas and brightly lit outdoor areas. One example is a surveillance camera shot of the truck driven by convicted Oklahoma City bomber Timothy McVeigh [Lin06]. The camera was in the lobby of a building with a view of the outdoor scene. The truck is barely visible in the brightly lit street; nonetheless, this image was used as evidence in trial. Outdoor lighting conditions can also change in the matter of a few seconds when winds cause clouds to move quickly.

Judder is perceived jumpiness of motion caused by frame rates that are low relative to the rate of motion. Judder may be caused either by subject motion or by

panning. For many years, cinematographers used rules to determine the maximum rate at which they could pan based on exposure and subject motion characteristics. However, now audiences have learned to use judder as a cue for fast motion [B16].

3.10 Stereo and Multicamera Systems

Stereoscopic cameras are sometimes used for art and entertainment; they are also used for computer vision. The distance between cameras—the *binocular distance*—is a key parameter; a larger binocular distance results in greater depth perception. 3D cinema was popular in the 1950s with films such as *Creature of the Black Lagoon*; viewers often reported headaches while watching these films. Studies later found that headaches were the result of editing that jumped between subjects of widely varying distances, resulting in rapid shifts of the eyes and muscle strain. While some modern 3D movies are shot stereoscopically, many are shot in 2D and post-converted into 3D using a combination of algorithms and artists. Postconversion is popular because 3D cinema cameras are both unwieldy and expensive.

A key operation for stereoscopic imagery is *disparity analysis* or *stereo correspondence*. As illustrated in Fig. 3.57, the different positions of the left and right eyes or cameras causes objects in the scene to appear in different locations in the left and right images; disparity varies with distance to the object. Because different objects are at different positions and give different disparity values, we need to compute disparity for each pixel in the image—disparity is a property of each pixel, not of the entire image. A simple approach to computing disparity d is to correlate the left and right images using an approach similar to motion estimation and use sum-of-absolute differences to judge the quality of the match at each possible disparity. However, this brute force approach is computationally expensive; simple algorithms to identify the best match may also introduce noise in the disparity map. Kosov et al. [Kos09] formulated the disparity problem using gray-level constancy and smoothness constraints; they efficiently solved for disparity using multigrid solving algorithms as well as an adaptive multi-level grid. Hirschmuller [Hir08] introduced semiglobal matching, which uses pixelwise matching as well as a smoothness constraint. Disparity is calculated hierarchically, starting with a low-resolution image, in order to provide estimates of the disparity for the cost function. Tombardi et al. [Tom08] compared cost aggregation methods for stereo correspondence. Ttofis et al. [Tto15] designed a hardware disparity engine that uses the Census transform for correlation. The Census transform [Zab94] encodes the relative brightness of the eight pixels adjacent to a given pixel. We compute a window of the adjacent pixels such that $window(i,j)$ is 0 if $I(i,j) < I(center)$ and 1 otherwise. We then convert the window values to an unsigned integer by

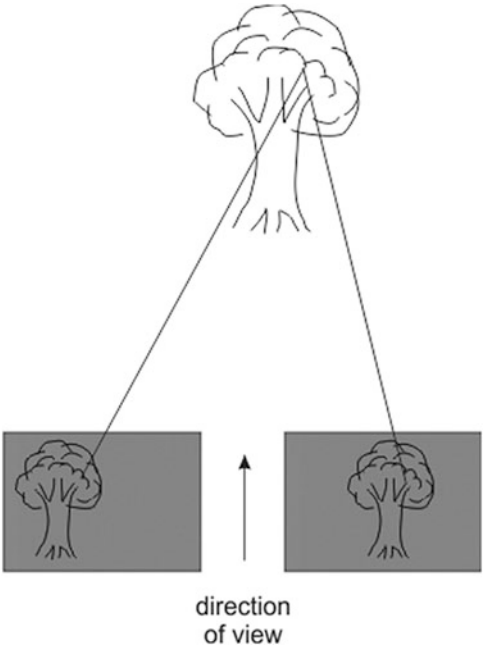


Fig. 3.57 Disparity in stereo vision

transposing them row by row, top down, and left to right, into the unsigned integer from MSB to LSB. The resulting value is the transform value.

Camera systems with more than two lens/image sensor systems are useful for a variety of purposes but are particularly useful for virtual reality capture, for which a very wide angle of view must be captured with as little geometric distortion as possible. To provide seamless global views, the cameras must be coordinated. Their shutters must be synchronized to avoid motion aliasing; global shutters should also be used rather than rolling shutters. A single exposure value is unlikely to be possible for many complex scenes—some of the cameras will see underexposed images, while others will see overexposed views. The images must be processed to blend their exposures and avoid block boundaries. These blending algorithms are a form of high-dynamic range imaging which we will discuss in Sect. 4.4.

3.11 Trade-Offs Revisited

As we noted at the start of the chapter, the physical world rarely presents us with win-win situations. We generally need to trade off reductions in one desirable characteristic in order to gain improvements in another. Now that we better understand the camera imaging chain, we can evaluate some of the trade-offs presented to us in digital camera design.

Larger image sensors have a clear cost. A larger image sensor requires a larger image circle, which in turn requires a longer focal length lens for any given angle of view. The larger optical system increases the size of the camera along the optical axis; the larger image sensor increases the camera size in the other two dimensions. (Larger image sensors also cost more to manufacture.) Let us call this relationship the *smartphone dilemma*—how do we build a camera that gives good images but is still physically small? We will see in Chap. 4 that algorithms can help us: high-dynamic range imaging can reduce the effects of physical noise; hyperresolution can reduce the effects of pixelization.

We need to determine whether our camera is limited by its optics or by its image sensor. Katz's formula from Sect. 2.8.5 suggested that imaging system resolution depended on the sum of the inverse squares of the component resolutions. But we can identify more specific constraints. The optical resolution can be characterized, at least to a first order, by its circle of confusion. The pixel size of the image sensor limits its spatial resolution. We would like the circle of confusion to be small relative to the pixel, but we face diminishing returns. On the other hand, the very small pixel sizes of some advanced image sensors—for example, 50 Mpixel sensors for full-frame 35 mm—provide enough resolution that very high-quality lenses are required to pay justice to the sensor's resolution.

We have seen in Sect. 3.4.6 that dark current limits the low-light performance of the sensor. We also saw that pixel noise limits the maximum useful resolution of the sensor.

Both video and still cameras face limitations on the resolution-frame-rate product. The bandwidth of the system must be sufficient to handle the total data volume generated each second. Before compression, we can express the data volume directly in terms of pixels; after compression, we express it in terms of bits. Video systems often subsample chroma information; still cameras may do so as well. The chrominance vs. luminance bandwidth affects the total required bandwidth. Chrominance subsampling is one method to satisfy bandwidth limitations while still delivering a required frame rate.

Further Reading

Nakamura's book [Nak05] provides detailed discussions of image sensor design and associated image processing.

Chapter 4
Image and Video Enhancement

4.1 Introduction

This chapter considers algorithms to enhance photos and moving images. High-dynamic range algorithms, for example, generate a composite image from several images at different exposures. These algorithms are increasingly available on cameras, but they are not part of the traditional imaging chain. Many of the algorithms here require substantially more computation than was the case for the methods of Chap. 3. In the next chapter, we will take this development a step further to look at algorithms that do not produce images at all—they analyze images to produce succinct descriptions.

The algorithms in this chapter do speak to our central challenge of autoprevisualization. Previsualization is often associated with the Zone System for exposure and development, but it refers to all of the many decisions that a photographer must make:

- Tonal mapping
- Framing
- Focus
- Subject pose

Algorithms can help us with all these steps: histogram equalization and high-dynamic range for tonal mapping, mosaicing and perspective transformations for framing, superresolution and focus stacking for focus, and facial detection and analysis for subject pose. Digital photographers often refer to manipulation of images after capture as *development* in analogy to film photography. However, most of the operations we perform with digital editing tools are closer to printing in film photography.

We can also move traditional previsualization considerations to produce enhanced images that could not be made using film: high-dynamic range frees us from some constraints on exposure and lighting; superresolution allows us to

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5_4

squeeze improved acutance from simple cameras and imperfect images; mosaicing allows us to capture a composite image of a large area, then decide later which section is most interesting. Some of these algorithms are directly applicable to video; others like mosaicing can be used to generate picture elements for video.

We will move through enhancement algorithms from simple to more complex. The next section introduces a few algorithms that we will use in this chapter, including the Gaussian pyramid and interpolation. Section 4.3 looks at tonal mapping and color grading. Section 4.4 considers high-dynamic range imaging. Section 4.5 discusses sharpening and superresolution, while Sect. 4.6 discusses the introduction of bokeh. Section 4.7 studies lens corrections. Section 4.8 studies focus stacking. Section 4.9 considers keystone correction; that algorithm is useful in the mosaic composition algorithms discussed in Sect. 4.10. We discuss video stabilization in Sect. 4.11. Section 4.12 considers software design issues for these algorithms. Finally, Sect. 4.13 surveys these results to consider their implications for photography.

4.2 Useful Algorithms

The Gaussian pyramid [Bur83] is a multiscale representation of an image as illustrated in Fig. 4.1. We can generate a low-pass filtered version of an image a Gaussian smoothing function, for example,

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{4.1}$$

Given an image I_{l-1}, we use the smoothing function to produce I_i. We then find the prediction error $L_i = I_{i-1} - I_i$. We perform this operation recursively to generate the image pyramid.

Interpolation algorithms have several uses. They can be used to simply increase the pixel count of an image; since these algorithms do not combine multiple images, they do not increase the image information. They can also be used to regenerate the uniform pixel field after distortion operations, for example, during keystone correction and mosaic composition.

Several algorithms have been designed with varying perceptual effects. *Nearest neighbor interpolation* is the simplest approach. It copies pixels from the original image to supply the interpolated values. It has poor perceptual characteristics. A somewhat more sophisticated approach is *bilinear interpolation*, which uses linear interpolation on a 2×2 window, resulting in a quadratic formula for the interpolated pixel $I_l(x_b, y_b)$. Given four pixels from the original image $I_1(x_1, y_1), I_2(x_2, y_1)$, $I_3(x_2, y_2), I_4(x_1, y_2)$, we interpolate two values in x, one for the upper pair of pixels and the other for the lower pair:

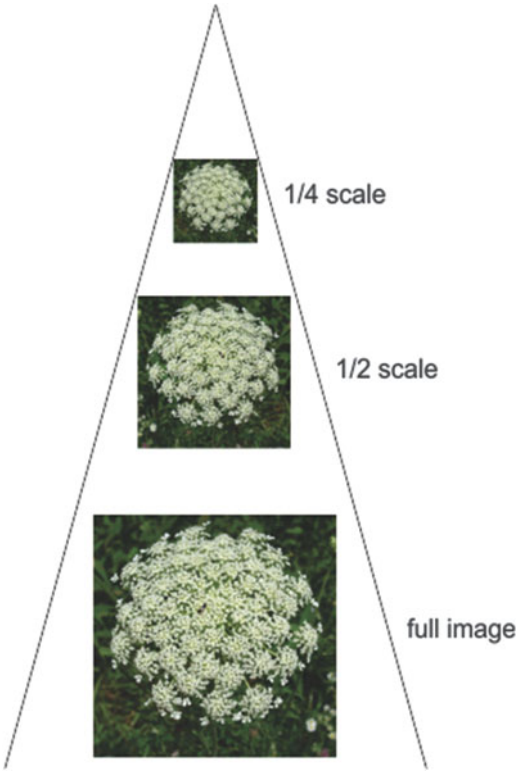


Fig. 4.1 A Gaussian pyramid

$$I_{x1} = \frac{x_2 - x_b}{x_2 - x_1} I_1 + \frac{x_b - x_1}{x_2 - x_1} I_2 \tag{4.2}$$

$$I_{x2} = \frac{x_2 - x_b}{x_2 - x_1} I_3 + \frac{x_b - x_1}{x_2 - x_1} I_4. \tag{4.3}$$

The final interpolation is performed in y:

$$I_l = \frac{y_2 - y_b}{y_2 - y_1} I_{x1} + \frac{y_b - y_1}{y_2 - y_1} I_{x2}. \tag{4.4}$$

Bicubic interpolation is widely used for upsampling and resampling images. It interpolates on a 4×4 window, resulting in a smoother image. Original image points are convolved with a kernel function to find the interpolated values. The kernel function is designed to be symmetric and continuous and to have a continuous first derivative; it is also designed to match the Taylor series expansion of the function as much as possible. The one-dimensional form of the kernel function is

$$u(x) = \begin{cases} 1.5|x|^3 - 2.5x^2 + 1 & 0 \le |x| \le 1 \\ -0.5|x|^3 - 2.5ax^2 + 4|x| + 2 & 1 \le |x| \le 2 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

The kernel is convolved with coefficients equal to the pixel values; the boundary values of the interpolation coefficients are $c_{-1} = c_2 - 3c_1 + 3c_0, c_{N+1} = 3c_N - 3c_{N-1} + c_{N-2}$. To interpolate, we take a 4×4 window of pixels in the original image I around $\langle j, k \rangle$ in the range $[-1, 2]$ in each dimension. The two-dimensional interpolation is performed by multiplying together the one-dimensional convolutions. We can interpolate a fractional-positioned point $\langle x, y \rangle, x, y \epsilon [0, 1]$ as

$$I_c(j + x, k + y) = \sum\nolimits_{-1 \le l \le 2} \sum\nolimits_{-1 \le m \le 2} I(j + l, k + m) u\big(x - x_{j+l}\big) u\big(y - y_{k+m}\big) \tag{4.6}$$

While bicubic interpolation has many desirable properties, the fact that kernel function assumes negative values at its outer edges means it does create some ringing undershoot.

Lanczos interpolation uses a sinc filter. The two-dimensional Lanczos window is

$$L(x, y) = \frac{a \sin \pi x \sin \frac{\pi x}{a}}{\pi^2 x^2} \frac{a \sin \pi y \sin \frac{\pi y}{a}}{\pi^2 y^2}, -a \le x, y \le a \text{ and } x, y \ne 0, L(0) = 1, 0 \text{ otherwise.} \tag{4.7}$$

A value is interpolated by discrete convolution.

Zhou et al. [Zho12] use interpolation to guide cubic interpolations. They first identify diagonal edges which they use to interpolate pixels on the diagonals between the original pixels. They then use horizontal and vertical edges to interpolate pixels above/below and left/right of the original pixels.

4.3 Tonal Mapping and Color Grading

The *tonal mapping* problem is a simple example of the types of operations we want to perform on images after capture. It also helps us better understand the role of post-capture operations in previsualization. Tonal mapping simply refers to how we map radiosities in the scene to gray levels in the final rendered image. A naïve view of photography puts all the responsibility for tonal mapping onto exposure. But the Zone System [Ada02B, Ada02C] teaches that decisions after exposure are critical to how the image is finally rendered.

The Zone System tells us to first choose a low-valued tone and a high-valued tone in the scene and decide in which zone each should be placed. In film, we then *expose for the low value and develop for the high value*. The exposure should be

chosen to ensure that the tone chosen for the lower zone is captured at that zone—a Zone III value, for example, would be captured by setting the exposure two stops below that required for Zone V middle gray. The film development time is then determined to place the higher zone. For example, if the higher tone is desired to be on Zone VIII but the exposure places it only on Zone VII, the film could be developed longer to make the required adjustment.

In film, we develop for the high tones because development changes the slope of the characteristic curve. Longer development affects the higher zones more than it does the lower zones—the lower zones stay at approximately the same density, while the higher zones become more dense, increasing the slope of the curve. Similarly, shorter development reduces the slope of the characteristic curve. We can therefore change the contrast in the image by changing development.

To achieve the same result in digital photography, we need to use digital tools. We can change the characteristic curve of an image by consistent relative adjustment of the pixel values. Figure 4.2 shows a before-and-after example of characteristic curve adjustment. In this case, the tool allows us to change the slope of the characteristic curve by moving its black and white endpoints; it then adjusts the pixel values to conform to the new curve. The resulting image has more contrast with brighter whites and deeper blacks, making it more pleasing to look at; we saw in Chap. 2 that the human visual system prefers scenes with strong black and white values so that the visual system can calibrate itself. To understand the transformation, let's start with a linear characteristic curve:

$$p_o = ap_i + b \tag{4.8}$$

The transformed output pixel p_o's value depends on the input pixel value $p_o \epsilon [0, W]$, the characteristic curve slope a, and its y intercept b. For the unmodified image, $a = 1, b = 0$. We can change both the slope and the intercept. Then the value of each pixel in the transformed image is

$$I_o(i,j) = aI_i(i,j) + b \tag{4.9}$$

with the understanding that I_o ranges over $[0, W]$. As with film, changes to the slope have the greatest effect on the higher zone values, although the slope does affect all the pixel values to some extent. We can change the mapping of the lower pixel values by changing the y intercept.

However, digital processes give us flexibility that we do not have in chemical photography. We can choose pretty much any characteristic curve we want and apply it to the image. Figure 4.3 gives an example of a nonlinear characteristic curve transformation which allows us to change the balance between light and dark areas in subtle but useful ways.

We change zone values in color images by applying the same characteristic curve transformation to all of the color channels. We can also change the color balance of the image by applying different transformations to different color channels. In Fig. 4.4, we can increase the yellow in the image by reducing the

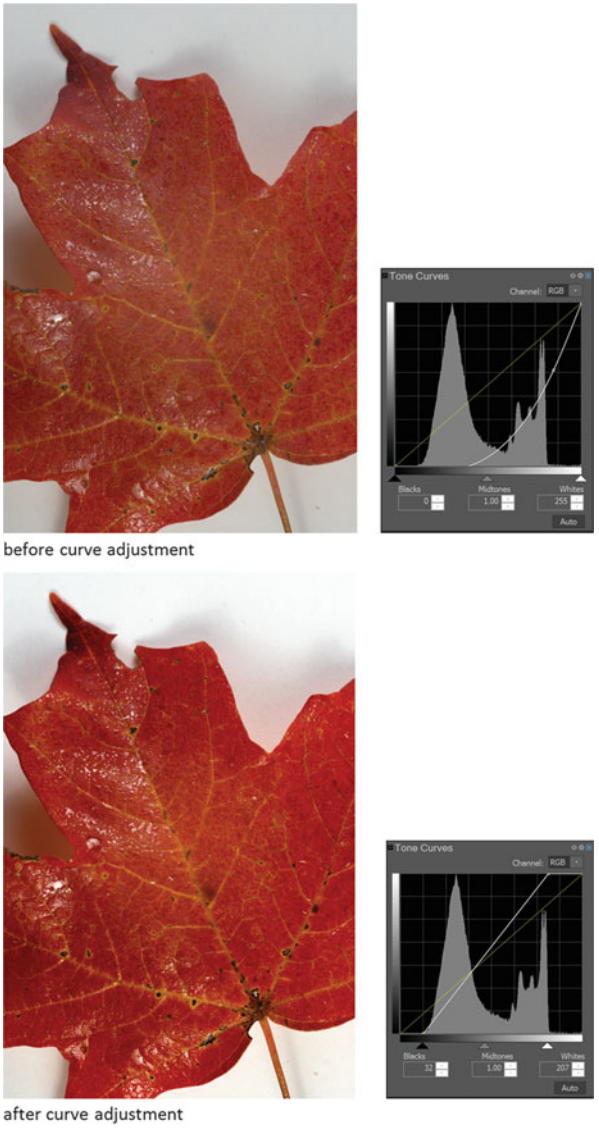


Fig. 4.2 Characteristic curve adjustment

blue; this curve makes the yellow most pronounced in the darker regions of the image.

Once again, we have more freedom than with film. Color film does not work well with the Zone System because different layers respond differently to changes in development time, resulting in color shifts.

Remember that tonal mapping of our digital images has fundamentally different goals than does color management. Both perform similar mappings of luminosities and color values. But the color manager's goal is to keep the image rendering

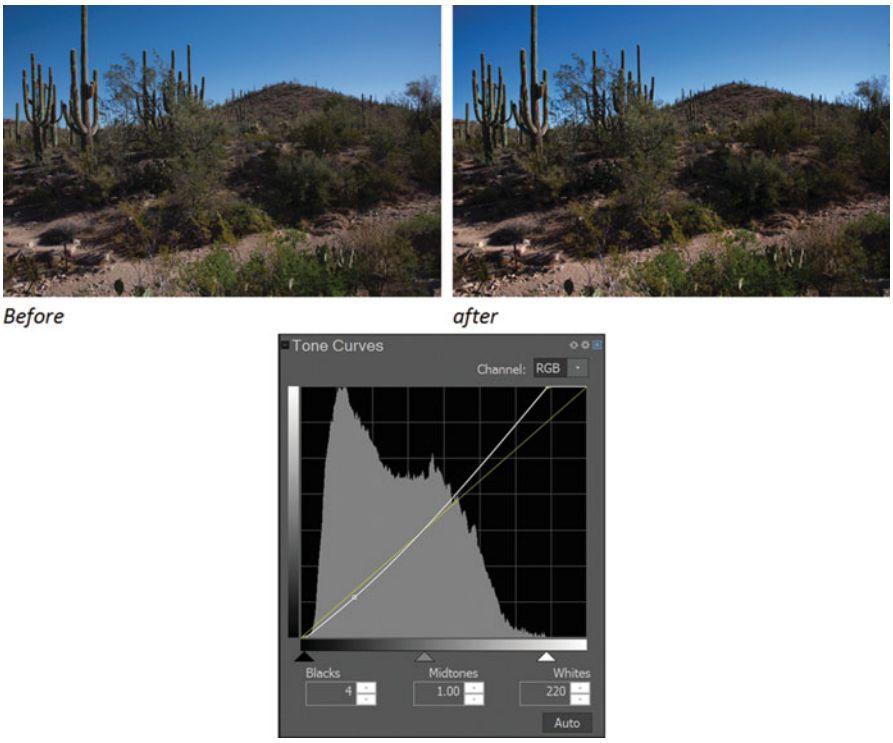


Fig. 4.3 Effects of applying a nonlinear characteristic curve transformation

consistent across displays that have different characteristics. Tonal management, in contrast, is designed to change how the image looks.

Histogram equalization is an algorithmic approach to tonal mapping; its underlying assumption is that pixel values should be uniformly distributed. This rule tends to give good, easy-to-read images.

We can treat the pixel values as a posterior probability distribution. Let's assume that we have W bins, one for each possible pixel value. If $n_b(k)$ is the number of pixels in bin k (the number of pixels whose intensity equals the bin value), then we can represent the probability density function of the pixels as

$$h(k) = \frac{n_b(k)}{N} \tag{4.10}$$

where N is the total number of pixels in the image. If we consider the pixel values to be real-valued over $[0, 1]$, the uniform distribution of pixel values would give a probability density function of $p_I(k) = 1$; the cumulative distribution function would be a line of slope 1 and y intercept at the origin. For our discrete histogram, the cumulative histogram function is

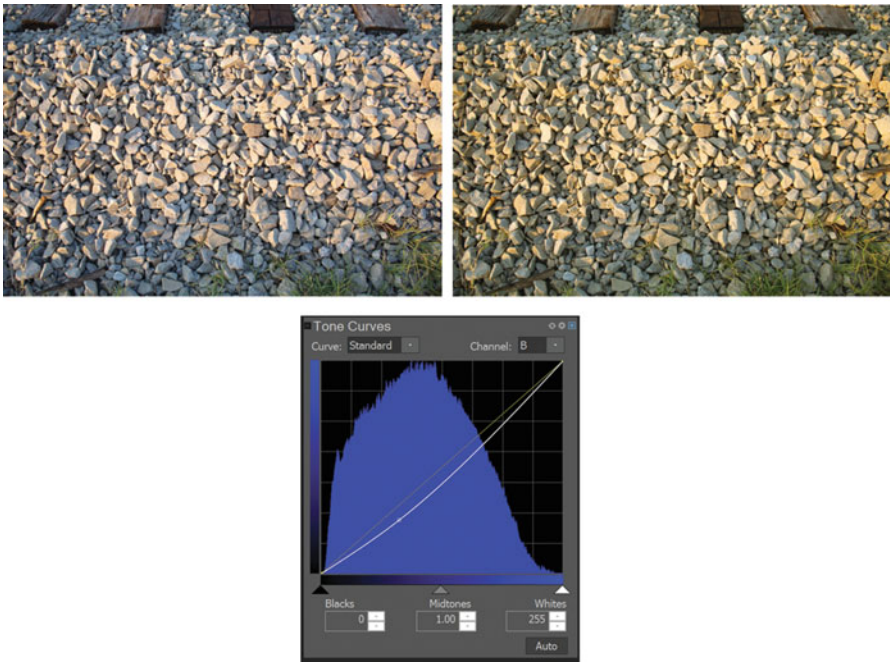


Fig. 4.4 Changing the color balance of an image with the characteristic curves

$$H(k) = \frac{1}{N} \sum_{0 \le l \le k} h(l). \tag{4.11}$$

This cumulative function gives us the mapping required from the image's pixel distribution to a uniform distribution.

The equalized image's pixel values are

$$I_o(i,j) = WH(I_i(i,j)). \tag{4.12}$$

The W factor translates the cumulative histogram to the range of pixel values.

Larson et al. [Lar97] developed a modified form of histogram equalization. They found that linear histogram equalization increased the contrast of low-contrast areas. They limited local contrast changes to be no more than the global contrast value:

$$h'(k) \le \frac{N\Delta h}{\log L_{\max} - \log L_{\min}} \tag{4.13}$$

where Δh is the size of a histogram bin and $[L_{\min}, L_{\max}]$ is the luminance range of the display or output image. They used an iterative algorithm to adjust the histogram.

Color grading is the term used for the image adjustment process in motion picture production. *Color timing* refers to the photochemical version of this process, which adjusted exposure times to control color. Digital workflows give filmmakers much finer-grained control over their images and introduces correspondingly more work. Colorists perform both corrective and artistic operations; they work both within an image and across shots. Basic corrections may include exposure, white balance, contrast, and noise. An early part of the process is to create one or more *lookup tables* (*LUTs*) to map pixel values from the input sequence to values in the print. Colorists work with the director and cinematographer to build lookup tables that reflect the color and lighting scheme for each part of the film; many films use different LUTs for different scenes or act to convey emotion through lighting. The LUT may be adjusted manually, one value at a time, to create the desired look; this level of detailed control goes far beyond what is possible with film. Many motion pictures shot on film are scanned so that color grading can be performed digitally.

4.4 High-Dynamic Range Images

High-dynamic range (*HDR*) imagery combines several images to create a merged image that displays more clearly the range of illumination in the scene. Most algorithms use different exposures—*bracketing* the exposure—but we will see one algorithm that combines a burst of several images taken at the same exposure. An example is shown in Fig. 4.5. Creating an HDR image requires us to do two things: determine how the scene luminance of each point in the image is determined from the set of images and map from the scene luminance to the display luminance to compresses the dynamic range onto the display's limited range. We have a variety of criteria with which to compress the expanded dynamic range onto the display range, depending on our model of what is important to the viewer as well as the computational effort we are willing to expend.

An early approach to HDR was developed by Mann and Picard [Man95]. They weighted output pixel values $v \rightarrow v_0$ using the function

$$v_0 = \left[\frac{d}{dv}\log g(v)\right]^{-1} \tag{4.14}$$

where $g(v)$ is the camera response function.

Debevec and Malik [Deb97] used a peak-shaped weighting function that favors midrange pixel values:

image pair

high dynamic range image

Fig. 4.5 A high-dynamic range image and its component images

$$v_0 = \begin{cases} v - v_{\min}, v \leq \dfrac{1}{2}(v_{\min} + v_{\max}) \\ v_{\max} - v, v > \dfrac{1}{2}(v_{\min} + v_{\max}) \end{cases}. \tag{4.15}$$

In these formulas, $v_{\min}, v_{\max}$ are the smallest and largest pixel values in the image.

In addition to mapping tonal values, high-dynamic range algorithms can try to minimize noise. Early approaches concentrated on quantization noise. Later methods developed more detailed noise models of sensors. Granados et al. [Gra10] developed a noise model that included photon shot noise, dark current shot noise, readout noise, photoresponse nonuniformity, and dark current nonuniformity. They developed a weighting function to minimize the variance of the luminance reconstruction:

$$w_{\text{opt}}(v) = \frac{t_i^2 g^2 a_j^2}{g^2 t_i \left(a_j \mu_X + 2\mu_D\right) + 2\sigma_R^2}. \tag{4.16}$$

In this formula, t_i is the exposure time, g is the gain of the complete imaging chain, a_j is the pixel gain, μ_X is the mean pixel value, μ_D is mean dark current, and σ_R is the readout noise including quantization error. Hasinoff et al. [Has10] developed a model based on the fact that ISO scaling improves image signal-to-noise ratio, as we discussed in Section 3.sensor.analysis. They modeled SNR for a single image as

$$SNR(\Phi)^2 = \frac{\Phi^2 t^2 \left[I < \widehat{I_{\max}(g)}\right]}{\Phi t + 2\sigma_{\text{read}}^2 + \sigma_{ADC}^2 g^2} \tag{4.17}$$

where Φ is the irradiance, g is camera gain, t is exposure time, σ_{read}^2 is the read noise variance, and σ_{ADC}^2 is the quantization noise of the analog-digital converter. The function $\left[I < \widehat{I_{\max}(g)}\right]$ is a binary operator that enforces zero SNR for saturated pixels. They showed that under their model, the SNR of the merged set of images is linear in the size of the image set. They formulated the problem of finding the optimal set of exposures to minimize SNR as an integer program.

Hasinoff et al. [Has16] developed a burst-based approach to HDR that has been implemented in the Android Camera2 API. Unlike the other methods, their approach combines images of the same exposure; burst sets in size range between two to eight images. This approach works because the burst of constant-exposure images serves to integrate the luminances in the same way that a single long exposure would.

Their system operates directly on the raw image value without color filter array interpolation or other operations. To minimize the effects of shake, they align the images in the set using a four-level Gaussian pyramid; they then perform a subpixel alignment estimate. Their merging algorithm models pixel noise as shot noise; they estimate the noise on a tile-by-tile basis using the root-mean-square value of the tile's pixel values. They merge by computing the 2D DFTs $T_z(\omega)$ of each tile for in image $z\ \epsilon[0, N-1]$. They compute an averaged value as

$$\widehat{T_0}(\omega) = \frac{1}{N} \sum_{z=0}^{N-1} (1 - A_z) T_z(\omega) + A_z T_0(\omega). \tag{4.18}$$

The parameter A_z controls the blending between the alternate frame z and the reference frame:

$$A_z(\omega) = \frac{|T_0(\omega) - T_z(\omega)|^2}{|T_0(\omega) - T_z(\omega)|^2 + c\sigma^2}. \tag{4.19}$$

They map the dynamic range of the pixels onto the display using a weighted sum of three criteria [Mer10]: they use the absolute value of a Laplacian filter on the grayscale image as a metric for contrast; they compare the pixel value's R, G, and B channels to the standard deviation of the channel to evaluate saturation; they emphasize midrange values using a Gaussian curve to measure well-exposedness. Their weighting function for these three criteria has a power law form:

$$C_{ij,v}{}^{\omega_C} S_{ij,v}{}^{\omega_S} E_{ij,v}{}^{\omega_E}. \tag{4.20}$$

Their system also performs a number of other operations, including correcting for lens vignetting, white balancing, color correction, dehazing, chromatic aberration correction, and sharpening.

Virtual reality omnidirectional video capture requires simultaneous HDR processing of all the cameras that contribute to the VR stream. Popovic et al. [Pop14] developed an FPGA-based processing system to perform Debevic and Malik's HDR algorithm in real time on 16 cameras.

The effect of high-dynamic range processing on the viewing experience varies depending on the scene. Figure 4.6 shows two examples of HDR photos, both taken with Android cameras. The mountain sunrise photo shows detail in most sections of the image, but the foreground appears brighter than it seemed, giving a slightly surreal result. The airport sunset photo fairly accurately captures the experience of this very contrasty but beautiful scene.

The example of 4.5 is based on only two of the five bracketed images taken of the scene. HDR images generated from different combinations of those images resulted in different renderings of the scene, a result of both the varying data from the images and the effects of the HDR rendering algorithm. This rendering, based on the extremes of the exposure bracket, preserved the contrast between the shaft of light and the dark forest. Using only two images minimized the effect of movement of the leaves under the afternoon breeze.

4.5 Sharpening and Superresolution

The *unsharp mask* filter is widely used in post-processing to sharpen images. The name comes from the photographic practice of using a mask to burn in the area around edges; the mask is blurred to avoid creating lines at the edge of the mask. Despite its name, the unsharp mask is a linear filter. The principles are best illustrated in continuous form [Spr12]. The unsharp mask is given by the convolution of the image with a Gaussian:

$$U(x,y) = I(x,y) * \left[\frac{1}{\sigma\sqrt{2\pi}} e^{-\left(x^2+y^2\right)/2\sigma^2} \right]. \tag{4.21}$$

Fig. 4.6 Examples of high-dynamic range images

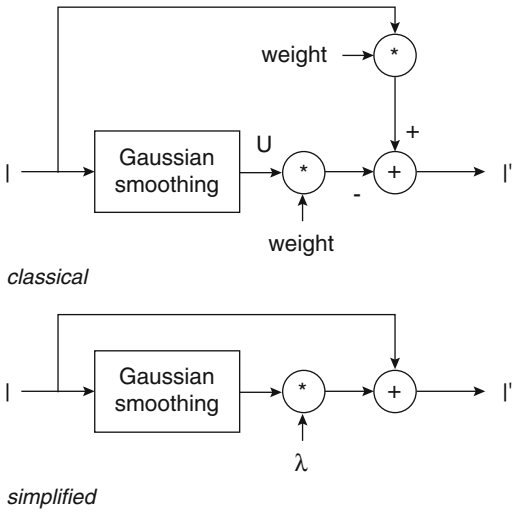


Fig. 4.7 The unsharp mask filter

where * is the convolution operator. The standard deviation gives the radius of the filter. As illustrated in Fig. 4.7, the sharpened image can be generated using a smoothing parameter c to weight the original and mask images:

$$I'(x,y) = \frac{c}{2c-1} I(x,y) - \frac{1-c}{2c-1} U(x,y). \tag{4.22}$$

A digital unsharp mask filter is built with discrete filters for the mask. The classical approach can be used with a Gaussian filter. A simpler implementation of this filter, also shown in Fig. 4.7, is to use a Laplacian filter or a Sobel operator as an edge detector, weight the mask by a factor λ, and then add it to the image.

Superresolution algorithms go beyond sharpening and upsampling to create an image with a higher pixel density than the original image. Superresolution has its artistic uses but is particularly useful in technical applications; for example, it is widely used for license plate readers. Four major approaches have been developed:

reconstruction from several lower-resolution images, interpolation, machine learning, and algorithms based on compressed sensing.

Tsai and Huang [Tsa84] formulated reconstruction from low-resolution images using Fourier transforms. They analyzed the properties of shifted component images in the Fourier domain and used the shifted component spectra to solve for the superresolution image. Their procedure required knowledge of the shifts between the component images. Hardie et al. [Har97] formulated the problem of finding the relative shifts of the components as a maximum a posteriori problem. They modeled the prior for the density of the superresolution image as a Gaussian; they included a value for each pixel and a 1/4 weight for each of its four cardinal (north, south, east, west) neighbors. They assumed translational motion. They used a gradient descent algorithm to solve for the MAP; they simultaneously solved for the superresolution image values and the translations between the component images. Patti et al. [Pat97] took into account motion blur during reconstruction. They used an integral model of motion blur similar to the model we saw in Sect. 3.4.6. They noted that the effect of subject motion is to perform a homographic transformation on the rectangular pixel. They use the method of projection onto convex sets to solve the estimation problem. At each step, they compute blur for each site and estimate an updated image; they then compute the residual and backproject it onto the component images, then apply a stopping criterion. Robinson et al. [Rob10] used an FFT implementation of a Weiner filter to denoise the merged superresolution image; they used spatially varying estimates of the noise variance. Some cameras capture multiple images for superresolution by shifting the sensor using piezoelectric actuators similar to those used for image stabilization. This approach also gives a full set of color filter array samples at each location, avoiding the need for CFA interpolation.

The RAISR algorithm [Mil16] uses machine learning for superresolution. The algorithm is rained on pairs of low-quality/high-quality images to recreate details similar to those in the high-quality version from features in the low-quality version. Training is performed on edge features based on direction, strength, and coherence (consistency of directionality). This approach was found to be comparable to Lanczos interpolation.

Hou and Andrews [Hou78] used cubic spline-based interpolation for superresolution. The form of the third-order spline is

$$\frac{1}{6\Delta^4}\Big[(\xi-\xi_{k-2})^3U(\xi-\xi_{k-2})-4(\xi-\xi_{k-1})^3U(\xi-\xi_{k-1})+6(\xi-\xi_k)^3U(\xi-\xi_k)$$
$$-4(\xi-\xi_{k+1})^3U(\xi-\xi_{k+1})+(\xi-\xi_{k+2})^3U(\xi-\xi_{k+2})U()\Big]. \qquad (4.\text{super}-\text{pline})$$

In this formula, Δ is the grid spacing and $U()$ is a unit step function. They used a digital filter to interpolate the curve given a sampling interval δ and a fixed multiple m:

$$\frac{1}{6m\delta}\left[\left(1+4z^{-1}+z^{-2}\right)/z^{-1}\right]\left[z^{2(m-1)}\right]\left[\left(1-z^{-m}\right)/\left(1-z^{-1}\right)\right]^4. \tag{4.23}$$

Machine learning methods use learning algorithms to infer the prior co-occurrences between the component images and superresolution image. Sun et al. [Sun03] decomposed a low-frequency image to identify and classify primitives. Off-line training had associated the low-frequency primitives with high-frequency primitives. A consistent set of high-frequency primitives was found using a Markov model. The high-frequency components were then combined with the low-frequency image to create the superresolution image.

Compressed sensing is widely used in modern signal processing; sparse signal models allow sampling rates below the Nyquist criterion to be used. Yang et al. [Yan10] used compressive sensing methods to create a superresolution image from a single low-resolution image. They used a training algorithm to identify relationships between low-resolution and high-resolution image patches. Their training algorithm used a common indexing scheme for the low- and high-resolution image patches; this allowed a low-resolution patch to be used to look up its corresponding high-resolution patch. Their dictionary, which was overcomplete, was based on a linear combination of a set of atomic features. The high-resolution patches are combined to create the superresolution image. They applied their approach both to general images and to images of faces; knowledge of the subject characteristics allows lower-resolution component images to be used.

4.6 Bokeh Introduction

Shallow depth of focus is often used to separate the subject of the photo from the background. Photographers have long exposed at wide apertures to create a shallow depth-of-field and render the background with bokeh. Lenses with shorter focal lengths also provide shallower depth-of-field which can be used to increase the bokeh of the background. However, smartphone lenses often operate at narrow apertures. Some cameras use stereo information, either from stereo analysis or a time-of-flight imager, to separate foreground and background regions. The background regions are then convolved with a kernel operator to create an out-of-focus background; a variety of bokeh effects can be created by proper choice of the blur kernel.

4.7 Lens Corrections

As we saw in Chap. 2, lenses introduce a wide range of distortions and aberrations. Correcting those problems after image capture requires a model of the lens. While in principle the optical characteristics of the lens could be derived from its design,

we rarely have enough information to do so reliably. We generally rely on basic models, often with parameters that are reverse engineered from camera calibration.

Fisheye distortion has received a great deal of attention. This geometric distortion becomes pronounced at focal lengths much shorter than the normal focal length for the sensor format. Hughes et al. [Hug08] survey algorithms for fisheye distortion. A simple radial distortion model [Len88] fits the distortion parameters to a second-order polynomial:

$$X_d = \frac{X_u}{1 + \kappa_1 R^2}, Y_d = \frac{Y_u}{1 + \kappa_2 R^2}, R^2 = X_u^2 + Y_u^2. \tag{4.24}$$

Lower-order models do not provide sufficient correction for very wide-angle lenses. Shah and Aggarwal [Sha94] proposed a fifth-order model with both odd and even terms:

$$\theta' = a\theta + b\theta^2 + c\theta^3 + d\theta^4 + e\theta^5, \tag{4.25}$$
$$\rho' = a\rho + b\rho^2 + c\rho^3 + d\rho^4 + e\rho^5, \tag{4.26}$$

where $\langle \rho, \theta \rangle$ is the position of the image point in polar coordinates and $\langle \rho', \theta' \rangle$ is its corrected position. An alternative model is the perspective model [Ish03]:

$$\rho' = \hat{f} \tan \frac{\rho}{\hat{f}}. \tag{4.27}$$

In this case, $\hat{f}$ is the apparent focal length of the fisheye, which may not be the same as its physical focal length.

This correction will, in the case of very wide fisheye lenses, leave some corrected image locations without pixels mapped to them. The missing pixel values can be filled in using interpolation.

Some camera and lens systems automatically capture lens data and record it in the image or video file. Data may include both the type of lens and its settings—focal length and aperture. This data can be used in post-processing to guide corrections. Databases of lens characteristics are available in both commercial and open-source versions.

4.8 Focus Stacking

Focus stacking combines several images of a subject to create a composite image with a larger depth-of-field. Figure 4.8 shows a pair of component images and the resulting photo-stacked image. Photostacking is particularly useful in macrophotography and microphotography where extremely small depths of field limit one's ability to resolve the subject without racking the focus.

As we saw in Chap. 2, focusing slightly changes the framing of the subject in the image as the lens moves relative to the image surface. This effect is often significant

image pair

focus stacked image

Fig. 4.8 Focus stacking

in situations that call for focus stacking. The component images need to be registered using homographies to properly register the component images.

Hariharan et al. [Har07] develop focal connectivity maps to generate a focus stack. They estimated focus by filtering each component image with x and y Sobel operators and then used the results to generate a sharpness map for each component image i:

$$S_i(x,y) = \sqrt{I^2_{x[i]}(x,y) + I^2_{y[i]}(x,y)}. \tag{4.28}$$

They low-pass filtered the sharpness maps to help to combine local focus regions. They then used a sharpness threshold test to generate the focus stack image from the component images.

Federov et al. [Fed06] broke the component images into tiles and applied a focus criterion to select which component image would represent each tile position. They then treated the tiles as mosaic elements and combined them using multi-resolution spline methods.

original image with keystone

corrected image with modified keystone

Fig. 4.9 Keystone correction

4.9 Keystone Correction

Keystone correction, shown in Fig. 4.9, is useful for both artistic and technical purposes. (The name comes from the shape of the keystone at the top of a stone arch.) It can be used to substitute for camera swings and tilts in photographs of buildings. Car backup cameras are an often pointed downward and their imagery

benefits from keystone correction. This operation transforms the four corners of the keystone to the corrected positions of the corners: $\langle p_1, p_2, p_3, p_4 \rangle \rightarrow \langle p'_1, p'_2, p'_3, p'_4 \rangle$. For each point $p_i = \langle x_i, y_i \rangle$, we can write (letting $h_{33} = 1$)

$$x'_i = \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}y_i + h_{32}y_i + 1}, y'_i = \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}y_i + h_{32}y_i + 1}. \tag{4.29}$$

We can rewrite these formulas by multiplying by the denominator to put full set of relationships between the four point pairs in matrix form:

$$X' = KH, \tag{4.30}$$

$$\begin{bmatrix} x'_1 \\ y'_1 \\ \vdots \\ x'_4 \\ y'_4 \end{bmatrix} = \begin{bmatrix} x_1 y_1 1\ 0\ 0\ 0 \left(-x_1 x'_1\right) \left(-y_1 x'_1\right) \\ 0\ 0\ 0\ x_1 y_1 1 \left(-x_1 y'_1\right) \left(-y_1 y'_1\right) \\ \vdots \\ x_4 y_4 1\ 0\ 0\ 0 \left(-x_4 x'_4\right) \left(-y_4 x'_4\right) \\ 0\ 0\ 0\ x_4 y_4 1 \left(-x_4 y'_4\right) \left(-y_4 y'_4\right) \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ \cdots \\ h_{31} \\ h_{32} \end{bmatrix}. \tag{4.31}$$

The homography parameters can be found from the positions of the four corresponding point pairs as $(K^T\ K)^{-1}(K^T\ X')$. Numerically solving for these parameters does require some care; Hartley [Har97] used a preconditioning method to minimize the effects of noise in the values of the points.

4.10 Mosaic Composition

Mosaicing creates a composite picture from several images with overlapping fields of view. (This procedure is very different from the *demosaicing* used to interpolate pixel values from color filter arrays.) Mosaicing relies on homographic projections to transform the component images. It has a number of artistic and technical uses. We can use mosaicing to build a larger image with more detail, change aspect ratio, or shoot now and crop later. Mosaics can also be used to build background for other applications and to create synthetic views such as video rearview mirrors. Huge panoramas can be made with dozens or hundreds of component images, allowing viewers to zoom in for detailed views. While these large mosaics could be thought of as superresolution, they are not constructed using subpixel interpolation or estimation techniques.

The first step in synthesizing a mosaic is to determine the geometric projection used from the scene to the synthetic image. As we saw in Sect. 2.5.4, we can choose between several different projections for panoramic scenes. Figure 4.10 shows three possible projections: perspective, cylindrical, and spherical. In many cases, the photographer has not been precise about camera movements, so without a detailed reconstruction of camera movement, any projection will be approximate. The rotations of the camera should be performed around the lens' nodal point to

Fig. 4.10 Projections of component images onto mosaics

avoid perspective shifts, but this effect becomes less significant as subject distance increases; it is rarely a consideration for distant landscapes.

Szeliski and Shum [Sze97] developed an algorithm for mosaicing that does not restrict the motion of the camera during image capture. Their method requires estimating the focal length of the lens, which in turn requires estimating a homography matrix between a pair of images I_0 I_1 (or possibly several pairs). They generate the homography parameters using an iterative method; at each

iteration, they warp the image I by $I+D$ where the update matrix D has $d_{33}=0$ and nonzero parameters otherwise. At a point x, y, this update can be written as

$$x'' = \frac{(1+d_{11})x + d_{12}y + d_{13}}{d_{31}x + d_{32}y + 1}, y'' = \frac{(1+d_{21})x + d_{22}y + d_{23}}{d_{31}x + d_{32}y + 1}. \tag{4.32}$$

They use a Jacobian matrix (the partial derivatives of the position x'' with respect to the update values d) to optimize the parameters; these partial derivatives describe the motion from one frame to the other:

$$J_d(x) = \frac{dx''}{dd} = \begin{bmatrix} x\ y\ 1\ 0\ 0\ 0 - x^2 - xy \\ 0\ 0\ 0\ x\ y\ 1 - xy - y^2 \end{bmatrix}^T. \tag{4.33}$$

We then minimize

$$\sum_i \left[g_i^T (J_d(x))^T d + e_i \right]^2 \tag{4.34}$$

where g_i^T is the image gradient $\nabla I_1(x')$.

Given the homography matrix that relates two images, we can estimate the focal length of the lens.

The relationship between the two images I_0, I_1 is $V_1 R V_0^{-1}$ (assuming for generality that each shot has its own viewing matrix): the inverse of the view of camera 0, rotation to the position of camera 1, and the view of camera 1. The homography can be represented in terms of this rotation as

$$\begin{bmatrix} r_{00} r_{01} r_{02} \\ r_{10} r_{11} r_{12} \\ r_{20}/f_1 r_{21}/f_1 r_{22} f_0/f_1 \end{bmatrix}. \tag{4.35}$$

The first focal length can be estimated as

$$f_0^2 = \sqrt{\frac{m_0^2 + m_1^2 - m_3^2 - m_4^2}{m_5^2 - m_6^2}} \text{ if } m_5 \neq m_2 \tag{4.36}$$

$$f_0^2 = -\frac{m_0 m_3 + m_1 m_4}{m_2 m_5} \text{ if } m_2 \neq 0 \textit{ and } m_5 \neq 0 \tag{4.37}$$

Motion between frames must be estimated more accurately than in the case of video compression because we do not have an error signal to correct any discrepancies. The camera moves from frame to frame rotationally; each frame i is modeled by $V_i R_i$, the viewing and rotation matrices. We want to find the rotation vector $[\omega_x \omega_y \omega_z]$ for each frame. We can add the rotation estimation to the iterative estimation of the homography matrix. A rotational update matrix can be used to find a Jacobean matrix for the minimization procedure:

$$J_{\Omega} = \frac{dx''}{dd}\frac{dd}{d\Omega} = \begin{bmatrix} -xy/f\, f + x^2/f - y \\ -f - y^2/f\, xy/f\, x \end{bmatrix}^T. \tag{4.38}$$

We can map the component images into a spherical mosaic. The pixels in the mosaic have polar coordinates $[\theta, \phi]$; each pixel coordsponds to a 3D position

$$[X\ Y\ Z] = [\cos\phi \sin\theta\ \ \sin\phi\ \ \cos\phi\cos\theta]. \tag{4.39}$$

A point p in 3D space is mapped onto an image k as $T_k V_k R_k p$. A similar approach works for cylindrical and perspective mappings.

Brown and Lowe[Bro03] used SIFT features to find correspondences between component images. We will discuss SIFT features in more detail in Section 5.alg.

The edges of the component images will overlap; we need to find a way to join together the images. Agarwala et al. [Aga04] proposed merging images at naturally occurring seams in the component images. Their method makes use of an algorithm by Kwatra et al. [Kwa03]. The pixels in the region of the overlap between two images are modeled as a mesh. Each edge between pixels s and t is assigned a matching quality weight. A simple form of the weight is

$$\|I_0(s) - I_1(s)\| + \|I_0(t) - I_1(t)\|. \tag{4.40}$$

A more sophisticated form of this weighting function takes into account the horizontal and vertical gradients of the images. The minimum-weighted cut that separates the mesh into two components corresponding to the two sides of the boundary gives the best seam between the two images. This cut set can be found using standard operations research algorithms [Hil01].

4.11 Video Stabilization

Stabilizing video is more complex than eliminating the effects of camera shaking in still images. Camera movement may be part of the design of the shot. In the case of artistic shots, we want to maintain the flow of the shot while reducing shake. In some technical applications, we may want to correct for all camera motion, for example, to facilitate tracking. We will first discuss optical flow and then go into stabilization algorithms proper.

4.11.1 Optical Flow

Optical flow is a detailed, local analysis of motion that produces a *motion vector* at each point. It can be computed for every pixel in the image, although many applications require only a subset of pixel locations to be analyzed. Optical flow

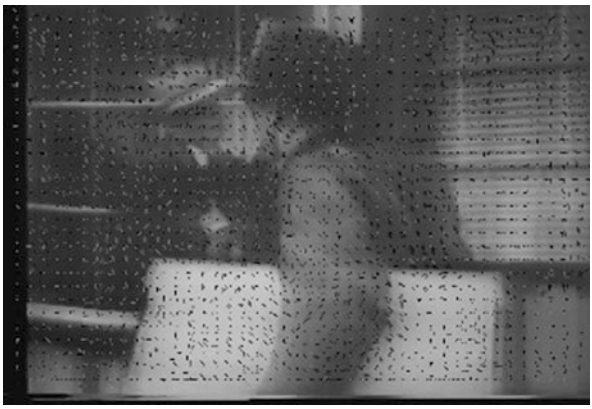


Fig. 4.11 Optical flow

is finer-grained than block motion estimation, which is designed to operate on relatively large macroblocks. Figure 4.11 shows an example of optical flow vectors computed for an image based relative to the previous frame in the video sequence. Optical flow is more computationally intensive than block motion estimation, but modern platforms can compute optical flow in real time.

The *Kanade-Lucas-Tomaso* (*KLT*) algorithm [Luc81, Tom91] is widely used to compute optical flow. It computes the gradient of luminance at each point with smoothness constraints. The optical flow at a point is computed on an $n \times n$ window; a 2×2 window is sufficient for simple applications, and larger windows give more accurate results, particularly with larger displacements. The partial derivatives of the frame I relative to its reference frame are I_x, I_y, I_t; these can be approximated using difference equations. The flow vector is given as

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \sum_{1\le i\le n}\sum_{1\le j\le n} I_x^2(i,j) & \sum_{1\le i\le n}\sum_{1\le j\le n} I_x(i,j)I_y(i,j) \\ \sum_{1\le i\le n}\sum_{1\le j\le n} I_x(i,j)I_y(i,j) & \sum_{1\le i\le n}\sum_{1\le j\le n} I_y^2(i,j) \end{bmatrix}^{-1} \begin{bmatrix} \sum_{1\le i\le n}\sum_{1\le j\le n} I_x(i,j)I_t(i,j) \\ \sum_{1\le i\le n}\sum_{1\le j\le n} I_y(i,j)I_t(i,j) \end{bmatrix}. \tag{4.41}$$

These formulas can be solved iteratively; a small number of iterations are typically required for small motion.

4.11.2 Stabilization Algorithms

Stabilizing a video signal requires that we adjust each frame so that objects do not move due to shake; the scene should move, however, as we deliberately move the

camera. The motion of the camera is known as *egomotion*. If the camera has accelerometers, the egomotion can be much more easily determined; if not, egomotion must be estimated from the video sequence. We must be able to separate egomotion from any subject motion in the scene.

A simple method [Mac93] is to use an oversized image sensor. As the lens moves relative to the image sensor, we select the appropriate part of the image sensor for use. If the image circle is smaller than the image sensor, we can use illumination directly to determine where the lens has moved due to shake.

The algorithm of Irani et al. [Ira94] selects an object on which it will base its motion estimation between two consecutive frames I_1, I_2. The displacement of an image point $[u\ v]$ representing scene point $[X\ Y\ Z]$, assuming small field of view and rotation, is

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -f_c\left(\frac{T_X}{Z} + \Omega_Y\right) + x\frac{T_Z}{Z} + y\Omega_Z - x^2\frac{\Omega_Y}{f_c} + xy\frac{\Omega_X}{f_c} \\ -f_c\left(\frac{T_Y}{Z} + \Omega_X\right) + x\Omega_Z + y\frac{T_Z}{Z} - x^2\frac{\Omega_Y}{f_c} + y^2\frac{\Omega_X}{f_c} \end{bmatrix}. \tag{4.42}$$

These equations follow from the perspective projection model of Sect. 2.6.1. They can be substituted into the formula defining a plane in 3D, $Z = A + Bx + CY$, to give eight parameters that describe 3D motion of a planar surface; Irani et al. refer to this form as a *pseudo-2D projective transformation*. The region with the dominant 2D motion is selected using a three-step process to gradually improve the selection: first translation using two parameters, then affine with six parameters, and then a pseudo-2D projective transformation with eight parameters. Estimation of the parameters is performed iteratively using a Gaussian pyramid. The resulting parameters are used to register I_1 and I_2. The registration process cancels the rotation component. Based on this result, the 3D translation between the two frames is computed. They then use the 2D motion parameters and 3D translation parameters to compute the 3D rotation of the camera.

We will consider camera motion again as part of tracking in Sect. 5.5.

4.12 Software Design for Image Enhancement

Numeric dynamic range determines the number formats we need to use for a computation. The most basic choice is between integer and floating point, but we can make further distinctions within each category. Modern processors often support efficient integer operations down to 8 bits of precision and up to 32 or 64 bits. IEEE floating point (IEEE 754) specifies formats ranging in length from 16 to 256 bits; many GPUs also support several floating point standards of varying lengths based on subtle variations to IEEE 754 which result in significantly smaller logic implementations. In purely software implementations, different number representations result in different run times and power consumptions. Custom

hardware designs provide us with an even wider range of choices with correspondingly larger tradeoffs between precision, performance, and power.

Given an arithmetic expression, we can evaluate the dynamic range required for the expression. We are given the dynamic range of each of the input variables. For simplicity, we can represent a variable in the form

$$\mp 0.nnn \mp ee. \tag{4.43}$$

In this description, the number of digits for the mantissa and exponent is to be determined. This representation does not restrict us to floating point formats. Keeping the number in a normalized form—the value is always shifted to eliminate leading zeros to the right of the decimal point—simplifies our analysis. We are primarily interested in the extreme ranges of the exponent, which will determine the number of bits required and help us decide whether to use a floating point representation.

Integer representations populate the number line uniformly. In contrast, floating point representations fill the number line at different densities for each different exponent value. The number line is less densely covered at larger exponents since the same number of mantissa values are spread across a larger range. In most cases, when a floating point representation is chosen, this varying range is not a major concern.

Addition and subtraction will at most change the exponent by 1. For example, $0.999\text{E}1 + 0.999\text{E}1 = 0.198\text{E}2$. Multiplication adds the exponent values and can double the size of the exponent. Division subtracts the exponent values, which in the case of a pair of negative exponents can result in the absolute value of the resulting exponent doubling. Given these rules, we can apply the operators from the expression to the dynamic range representations of the input variables to determine the dynamic range of each step. If we use the same number representation throughout the calculation, the required precision is determined by the worst-case dynamic range, not the dynamic range of the final result. KLT optical flow is an example of this principle [Sch15]. The input values are 8 bit (or perhaps 10 bit) pixel values; for most practical applications, the output dynamic range is even smaller given the limited range of a flow vector. However, the matrix inversion requires the larger dynamic range of floating point arithmetic. However, they found that reduced-precision floating point and a relatively simple division algorithm could be used given the required precision for image analysis.

We do not need to use the same number representation for the entire expression. However, conversion between formats imposes costs in execution time and power consumption.

4.13 Practical Image Enhancement

These enhancement algorithms allow cameras to provide novice users with better pictures than is possible using only the traditional image chain operations. A basic workflow for enhanced snapshots typically includes several elements:

- Face detection is used to identify faces which are then used to set the exposure and focus.
- Exposure settings can be analyzed to determine whether high-dynamic range processing is appropriate.
- White balance may be adjusted either by analyzing the scene automatically or from user inputs that classify the scene content.

Image enhancement algorithms offer the potential of more radical interventions: we can adjust composition, contrast, resolution, focus, perspective, framing, and, in the case of video, stability. Based on these algorithms, we have a great deal of freedom to create our photograph or video after shooting. Rather than carefully setting up the image and camera for a desired result, we can capture raw material and then adjust it at our leisure; we can make our final selections either immediately or after careful consideration.

Panoramas offer us a basic ability to shoot a scene but decide much later how to present it. Figure 4.12 shows a panorama with several possible croppings; these selections vary in their visual content and appeal. But panoramas do not allow us to move our point-of-view relative to the scene. We could image using a wandering video to later create the view we want from a 3D model of the scene.

We can also think of many operations that could be performed but are not commonly used today. For example, lighting adjustment is infrequently used, although algorithms do exist. This step could be used to change the flat or poor lighting of a person's face in a natural scene—perhaps to something more glamorous. We could go further and modify or synthesize the appearance and behavior of a person. A simple example would be to combine a smiling face with a different body pose. The commercial cinema has taken this process much further. *Rogue One: A Star Wars Story,* released in 2016, featured a performance by Peter Cushing, who passed away in 1994; it also featured a brief appearance by a youthful Carrie Fisher.

Fig. 4.12 Panoramas offer multiple cropping opportunities

Of course, with great power comes great responsibility. At some point manipulations stop being photography and move into the realm of animation. On the one hand, the prevalence of computer-generated imagery in film has probably shifted public appetites toward a slightly more synthetic style of photography. Highly stylized images have become very popular, suggesting that viewers have learned to enjoy and expect heightened reality. On the other hand, the uncanny valley probably exists for all sorts of images, not just people. Altered views of landscapes and other scenes may need to choose which side of the valley they reside: abstraction or representation.

Further Reading

Gonzalez and Woods [Gon17] provide a thorough introduction to image processing algorithms.

Chapter 5
Image and Video Analysis

5.1 Introduction

The human visual system is much more than a camera—most of the visual system is dedicated to analyzing the imagery captured by our eyes. We perceive the world both as scenic imagery and as our understanding of those scenes—people, objects, and places. Digital cameras have allowed us to move photography beyond imaging to image understanding. A camera does not need to take a picture—it can report on what it sees.

The results of our analysis depend entirely on the quality of the images upon which it is based. Every stage of the imaging chain affects the final result. Improper exposure, for example, can obscure an object of interest in the frame. In the case of video, variations in lighting across the scene can, if handled improperly, cause the subject's rendered appearance to change dramatically from frame to frame.

The algorithms of Chaps. 3 and 4 were pixel-intensive. While pixel-oriented computation requires large numbers of both arithmetic operations and data accesses, it has compensating advantages. Many pixel-oriented algorithms are fairly regular. Not only does regularity simplify programming, it also increases cache hit rates, resulting in higher performance and lower power consumption for the memory system.

The algorithms of this chapter move beyond pixel-oriented operations to extract features, identify objects and scenes, and analyze motion. As data becomes more abstract, the number of memory accesses decreases. But the number of operations per datum increases and the complexity/cost of those operations may also increase. Floating-point operations become much more common in later stages of analysis due to the wider dynamic range required. While floating-point addition and multiplication are not particularly costly—both can be executed in a single cycle by modern processors—floating-point division is inherently slower. Division is typically performed using iterative algorithms whose execution time may vary with data values.

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5_5

Higher-level analysis also requires more sophisticated data structures and access patterns on those data structures. While many pixel-oriented algorithms can operate on data structures of known and unchanging sizes, analysis often requires dynamic data structures. Dynamic memory management requires care to avoid *memory leaks* from data structures that are not unallocated once they are no longer needed. Passing varying amounts of data between tasks also requires *elastic buffers*.

Multicamera systems are often built on distributed computing platforms. Design of multicamera algorithms requires careful consideration of the abstractions to be passed between nodes in the distributed platform: too much data consumes too much bandwidth; too little data results in inadequate information for data fusion.

Section 5.alg introduces several important algorithms. Section 5.image.char considers low-level image characteristics. We then look at several important applications: video summarization in Section 5.video, visual search in Section 5. scene, and tracking and gesture recognition in Section 5.tracking. Section 5.multi generalizes some of these algorithms to multicamera systems. Section 5.apps briefly review the use cases and workflows made possible by these analysis algorithms.

5.2 Image Analysis Algorithms

The *Mahalanobis distance* is widely used to compare multidimensional data to a distribution. It measures the

$$M(\boldsymbol{x},\boldsymbol{\mu}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \tag{5.1}$$

where $\boldsymbol{x}$ is the measurement, $\boldsymbol{\mu}$ is the mean vector, and Σ is the covariance matrix. As we will see in Section 5.scene.id, this metric can also be used to compare two random vectors.

Corners are useful features which can be found efficiently. We are given a window function $w()$ to select a region of the function (the window can be a box or Gaussian) and a shift range $[U\ V]$ over which we test for the corner. The *Harris detector* is

$$H(i,j) = [i\ j] \left\{ \sum_{u \epsilon U} \sum_{v \epsilon V} w(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right\} \begin{bmatrix} i \\ j \end{bmatrix}. \tag{5.2}$$

SIFT [Low99, Low04, Low04B] is widely used to extract features from images; those images can then be compared with features extracted from other images. Figure 5.1 shows the correspondences between SIFT features in a pair of images, each of which takes a slightly different view of the scene. SIFT features are designed to be invariant to changes in scale, translation, and rotation; these features are also partially invariant to changes in illumination as well as affine or 3D

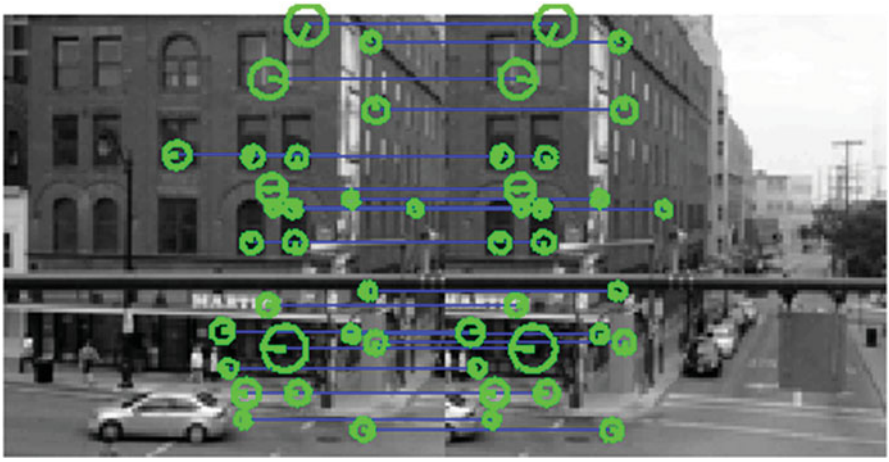


Fig. 5.1 Correspondences between SIFT features in two images

projection. Features are extracted using a set of difference-of-Gaussian versions of the image:

$$D(i,j,\sigma) = G(i,j,k\sigma)^{*}I(i,j) - G(i,j,\sigma)^{*}I(i,j) = L(i,j,k\sigma) - L(i,j,\sigma) \qquad (5.3)$$

where k is the scale factor applied to the image. The image pyramid is built starting from a $2X$ upsampled version of the image to allow high spatial frequencies to be analyzed. At a given level of the pyramid, 3×3 image patches are analyzed to find local minima and maxima. The locations of minima and maxima are compared, and only features which retain their identity at the adjacent scales are selected. To improve stability and matching, the extremal position $\hat{x}$ can be interpolated using a Taylor expansion of $D()$, with that interpolated position then used to refine the estimate of $D()$:

$$D(\hat{x}) = D + \frac{1}{2}\frac{(dD^{\mathrm{T}})}{dx}\hat{x}D \qquad (5.4)$$

Extrema with low values of $D(\hat{x})$ are rejected. A separate test rejects features produced by edges for which the principal curvature across the edge is large but small in the perpendicular direction. Given a Hessian matrix $\boldsymbol{H}$ of the second-order derivatives of $D()$, the required test is

$$\frac{\mathrm{Tr}(\boldsymbol{H})}{\mathrm{Det}(\boldsymbol{H})} < \frac{(r+1)^2}{r} \qquad (5.5)$$

where $\mathrm{Tr}(\boldsymbol{H})$ is the trace (sum of the eigenvalues) of $\boldsymbol{H}$, $\mathrm{Det}(\boldsymbol{H})$ is its determinant, and $r = 10$.

The reference orientation of the feature is determined from a histogram of local image gradient orientations, with the reference orientation chosen at the local maximum. The gradient magnitude and orientation can be computed as

$$m(i,j) = \sqrt{\Big((L(i+1,j) - L(i-1,j))^2 + (L(i,j+1) - L(i,j-1))^2\Big)}, \quad (5.6)$$

$$\theta(i,j) = \tan^{-1}\left\{\frac{L(i,j+1) - L(i,j-1)}{L(i+1,j) - L(i-1,j)}\right\}. \quad (5.7)$$

The feature vector is described as a 4×4 array of image gradient histograms from the region around the feature, each with 8 orientation bins, giving a total of 128 entries. This vector is normalized to unit length to reduce its sensitivity to illumination.

SURF [Bay06] extracts high-quality features with considerably less computational effort than is required for SIFT. SURF uses box filters as an approximation for Gaussians. The SURF descriptor is constructed by first assigning an orientation based on x and y Haar wavelet responses. A square region is defined by the assigned orientation, split into 4×4 subregions. The Haar wavelet responses in the dimensions defined by the orientation and the results from the subregions are summed; the sum of the absolute values of the results are also summed. The values are scaled to a unit vector to provide the descriptor components.

Mikolajczyk et al. [Mik05] compared region detectors that are covariant relative to affine transformations. Since affine transformations can be used to model changes in viewpoint, this class of detectors can be used to identify features across multiple cameras. They studied five detectors: Harris-Affine, Hessian-Affine, maximally stable extremal region (MSER), edge based, and intensity extrema based. The Hessian autocorrelation matrix is

$$H = H(\boldsymbol{x}, \sigma_D) = \begin{bmatrix} I_{xx}(\boldsymbol{x}, \sigma_D) & I_{xy}(\boldsymbol{x}, \sigma_D) \\ I_{xy}(\boldsymbol{x}, \sigma_D) & I_{yy}(\boldsymbol{x}, \sigma_D) \end{bmatrix} \quad (5.8)$$

The scales of selected features can be normalized for comparison. For either the Harris or Hessian detectors, an iterative region estimation algorithm can be used: first detect an initial region and select the scale; estimate the shape using the second moment matrix; normalize the region to be circular; re-estimate the shape if the eigenvalues of the second moment matrix differ.

RANSAC [Fis81] is widely used to fit models to data which contains a certain number of points with very large errors—a common situation in feature detection, for example. RANSAC operates iteratively. At each step, a subset S of data points is selected and the model is applied to those points. The result is a set of points $\hat{S} \epsilon S$ that fit the model within some error tolerance. If the size of $\hat{S}$ is above a threshold, it is used as the basis to generate a new S; if not, the next S is selected from scratch. The procedure terminates when either a consensus set has been found or the maximum number of allowed iterations has been reached.

Particle filtering [Dou08] is widely used to approximate problems whose underlying characteristics do not adhere to traditional mathematical assumptions, such as Gaussian distributions. We are interested in estimating a posterior density $p(x_{1:n}|\, y_{1:k})$ which describes an underlying sequence $x_{1:n}$ as represented by a sample sequence $y_{1:n}$. An importance density $q()$ allows us to draw samples that approximate the underlying distribution. A weight function $w()$ is used to shape the importance density toward the posterior density. Sequential importance sampling makes use of an importance density for which the sequence importance is the product of the members:

$$q_n(x_{0:k}) = q_1(x_1) \prod_{2 \le k \le n} q(x_k | x_{1:k-1}). \tag{5.9}$$

A sequence can be created sequentially with the new weight being computed from the previous weights and an incremental importance weight function. *Resampling* allows new samples to be generated from the previously estimated distribution. A *sequential Monte Carlo* procedure repeatedly generates a new sample for the sequence, computes the weights, and resamples until some stopping criterion is reached.

5.3 Image and Video Characteristics

This section looks at image and video characteristics from several perspectives. First, we consider the statistics of small image patches. Next, we consider algorithms to compute metrics that approximate psychovisual saliency. We then consider the selection of key frames in video sequences.

5.3.1 Image Statistics

Huang et al. [Hua00] studied the statistics of range images. They proposed a $1/r^2$ form for the distribution of single pixels. They found that a bivariate distribution model was a better fit for range images than for optical images. Lee et al. [Lee03] studied the statistics of both optical and range images. They concentrated on 3×3 patches. They found that optical images were dominated by two-dimensional features that coorespond to edges subject to camera blur. They found that range patches, in contrast, were grouped into a number of small clusters.

Zontak and Irani [Zon11] analyzed the distribution of features within an image; their 5×5 patches were somewhat larger than the 3×3 patches used by Lee, Huang, and Mumford. They compared the frequency of reoccurrence of image patches in a set of 300 images. They used the Parzen estimator [Par62] to estimate the empirical density of an image patch p within a neighborhood N

$$\text{density}(p;\text{dist}) = \frac{1}{\text{area}(N)} \sum\nolimits_{p_i \epsilon N} K_{\text{h}}\left(||p - p_i||^2\right) \tag{5.10}$$

where $K_{\text{h}}()$ is a Gaussian kernel. They found that the required radius for nearest-neighbor search grows exponential with the gradient of the patch:

$$\text{dist}(|\text{grad}|) = \beta_1 + \beta_2 e^{(|\text{grad}|/10)}. \tag{5.11}$$

They found that patches are repeated much more frequently within an image than between images.

5.3.2 Saliency

Saliency refers to the attention paid to a part of an image; as we saw in Chap. 2, the eye constantly scans a scene but may pay more attention to some areas than others.

Itti et al. [Itt98] developed a feature-based model that produces a *saliency map* of an image. Their approach is illustrated in Fig. 5.2. The feature operations are performed at nine different scales based on Gaussian pyramids. The feature extractors perform linear center-surround operations as the difference between pixels at two different scales.

A simple model of image statistics is based on the amplitude of the Fourier transform of the image, which is proportional to $1/f$ where f is frequency.

Hou and Zhang [Hou07] analyzed saliency using its spectral residual. They use as a metric the logarithm of the amplitude of the Fourier transform of the image:

$$\mathcal{L}(f) = \log A(f). \tag{5.12}$$

They find an average spectrum of the image as

$$A(f) = \frac{1}{n^2} \begin{bmatrix} 1 & 1 & \cdots \\ 1 & 1 & \ddots \\ 1 & \vdots & \ddots \end{bmatrix}^* \mathcal{L}(f). \tag{5.13}$$

The residual spectrum is

$$R(f) - \mathcal{L}(f) - A(f). \tag{5.14}$$

They take the inverse Fourier transform of the residual to create a saliency map.

Goferman et al. [Gof12] developed a context-aware saliency model based on a combination of low-level features, unusual global features, visual organization rules, and priors on object location. They measured local dissimilarity between two image patches as

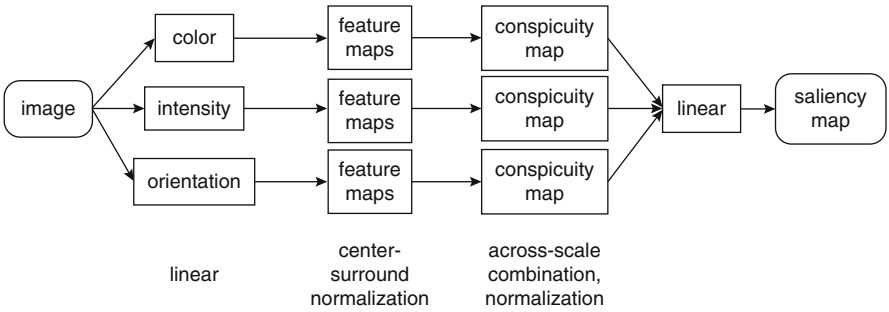


Fig. 5.2 Center-surround-based saliency mapping [Itt98]

$$d(p_i, q_i) = \frac{d_{\text{color}}(p_i, q_i)}{1 + c.d_{\text{position}}(p_i, q_i)}. \tag{5.15}$$

They used the constant $c = 3$; they defined color in the CIE L*a*b space. This results in a single scale saliency of

$$S_i^r = 1 - \exp\left\{ -\frac{1}{K} \sum\nolimits_{1 \le k \le K} d\left(p_i^r, q_i^r\right) \right\} \tag{5.16}$$

where K is the maximum saliency value. The saliency of a pixel i is its mean saliency at different scales:

$$\bar{S}_i = \frac{1}{M} \sum \text{r}\epsilon\text{R}\, S_i^r \tag{5.17}$$

where R is the set of patch sizes in the multiscale analysis. Context is modeled by weighting a pixel relative to its Euclidean distance $d_{foci}^r(i)$ to its closest attended pixel at scale r:

$$\widehat{S}_i = \frac{1}{M} \sum\nolimits_{r \epsilon R} S_i^r \left(1 - d_{foci}^r(i)\right). \tag{5.18}$$

They assumed that the subject is centered in the image and so multiplied $\widehat{S}_i$ by a centered Gaussian distribution; other models of the subject prior could be used.

Liu et al. [Liu11] used a supervised learning algorithm to create a saliency map. Their learning procedure estimated the weights of a set of features using the maximum-likelihood criterion. Features are combined linearly. They used multiple features for static images: multiscale contrast, center-surround histogram, and the spatial variance of color. They also considered saliency in video. They weighted the motion field as the exponent of the variance of the motion vector magnitude at each point. They considered several features of this weighted motion field: multiscale contrast, center-surround histogram, and spatial distribution. They also identified

coherence of moving objects by identifying whether a given pixel's color changed substantially from one frame to the next.

5.3.3 Key Frame Selection

Key frames can be selected by motion analysis [Wol96]—frames at the local minima of motion are selected at key frames. This approach is based on the observation that people tend to remember the stillest points in an action. This approach automatically determines both the number of key frames and the position of those key frames in the shot. It also takes into account both subject and camera motion.

Motion is estimated using the sum of the magnitudes of the optical flow vectors at each pixel:

$$M(t) = \sum_i \sum_j |o_x(i,j,t)| + |o_y(i,j,t)|. \tag{5.19}$$

Local minima can be determined by first identifying pairs of local maxima. After the first local maximum m_1 is selected, the next local maximum m_2 is chosen at the next point that varies by at least $n\%$ of the motion value for m_1. The local minimum between these two points is chosen as a key frame and m_2 is made to be the next m_1.

Figure 5.3 shows the 19 key frames selected by this algorithm from the resignation speech of President Richard M. Nixon; the frames also show the optical flow vectors. The source tape contained a flaw which caused a momentary disruption and the generation of some extra key frames. The speech lasted for 15 minutes and was televised as a single shot with no cuts. However, the camera operator zoomed in at several points during the speech. After having started with a wide shot of Nixon at the President's desk, it ended on a tight shot of his head. The zoom transitions are fairly subtle, but their cumulative effect is strong, particularly when viewed in this summarized format.

Alfred Hitckcock's *Rope* is a feature-length film in which every shot lasts the duration of a 10-minute reel of film. This film represents an extreme case of the need for key frame selection as a means of summarization. Early film cameras were heavy and bulky, resulting in static shots separated by cuts, dissolves, etc. A variety of cinematic innovations, including zoom lenses and Steadicam, have given filmmakers more freedom to reframe the shot rather than cut in order to control the viewer's point of view.

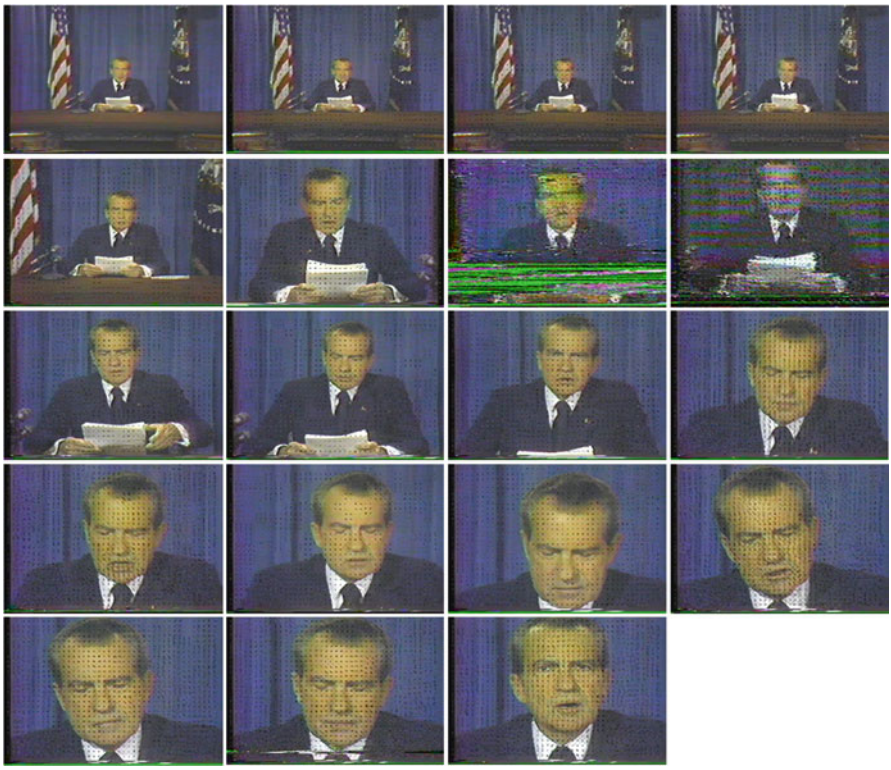


Fig. 5.3 Key frames automatically selected from the Nixon resignation speech

5.4 Scene Analysis

Scene characterization has a number of uses: casual photography may use scene characteristics to guide exposure and focus decisions; characteristics may also be used to search within large set of videos. We will first consider retrieval-oriented methods and then study face detection and recognition in more detail.

5.4.1 Visual Search

The QBIC system [Fli95] was an early and influential visual search system. QBIC was designed for use in visual databases. Images and videos were analyzed as they were loaded into the database; a query interface allowed users to formulate queries, which were then satisfied by a match engine. Object characteristics included texture, color, location, and shape. Scene characteristics included texture, color, texture and color as a function of position, and visual sketches. Video was

segmented into key frames and characterized by object and camera motion. The query languages allowed a query to be formulated in terms of these primitives. The match engine applied distance functions between the query and image descriptors. The traditional data structures for organizing searching in text are not always appropriate to the higher-dimensionality descriptors of video. QBIC used R* trees for low-dimensional features; it used principal component analysis to reduce the dimensionality of higher-dimensional features.

Sivic and Zisserman [Siv03] developed image descriptors for video sequences that allowed them to use text retrieval methods on images—the matches on descriptors can be precomputed. They construct two types of regions for each frame. One is constructed as an ellipse around an interest point by iteratively maximizing the isotropy of the intensity gradient in the region. The other type is identified as a region whose area is roughly stationary as the intensity quantization threshold is changed. Each region is represented by a 128-dimensional SIFT descriptor. They compare regions among adjacent frames and reject any region that is not stable for three frames. Regions are then clustered to reduce the dimensionality of the search space. The Mahalanobis distance is used to define the distance between regions.

Philbin et al. [Phi07] developed a set of techniques for search over large image databases. For each image, they extracted affine-invariant Hessian regions and generated a 128-dimensional SIFT descriptor for each one. They modeled both the images and search queries as sparse vectors of the occurrences of these descriptors. They used approximate k-means (AKM) to cluster the descriptors. Their clustering algorithm made use of eight k-dimensional (k-d) trees. The dimension on which to split each tree is chosen randomly from a set of dimensions with high variance; the split point is chosen close to the medium. The forest of trees creates overlapping partitions which help to control the curse of dimensionality. They use a combined priority queue for all trees to manage the search for a good partition for a given data point; the search is stopped at a fixed limit of paths. After initial search using approximate k-means, they rerank the initial set of results to take into account spatial information. They used a version of RANSAC to generate transformations that describe the spatial relationships between regions. They are able to test only a single pair of corresponding features for each image pair by making use of shape information to reject unlikely pairs and by restricting the set of transformations considered.

5.4.2 Face Detection and Recognition

Both face detection and recognition are important in many different applications [Zha03]. Face detection is useful in itself to identify the presence of people; it can also be used as a precursor to face recognition.

Leung et al. [Leu95] used a model-based approach combined with bottom-up features. They filter an image with an image pyramid of Gaussian derivative filters

and then match against template vectors to identify local feature matches. They use a graph to model constellations of features. They form a random vector X of normalized distances between features which they then normalize for length; they represent the normalized vector by its mean and covariance matrix. They use the maximum likelihood approach to estimate the scale of the face in the image. They search for faces by first identifying features with strong matches and then using the constellation graph vector to identify candidate positions for the missing features. They use an optimal discriminant to rank the candidate face positions.

Zhu et al. [Zhu00] used wavelet features to detect faces. They found a small set of wavelet features for faces based on training from a dataset that combined both faces and non-face images. They used a log-likelihood ratio test to classify a set of features derived from an image patch as being face or non-face.

The *eigenface* method [Tur91] can be used for both face detection and recognition. Recognition is based on Facebook of images $F = \{I_1, \cdots, I_M\}$. If each image is $N \times N$, it can be interpreted as an N^2-dimension vector, with the pixel value at each position giving the ordinate along the axis which represents that pixel. The set of pixels is analyzed using principal component analysis to fit the vector set into M components, each described by an orthonormal vector u_i and eigenvector λ_i. Any given face can then, in principle, be described as a linear combination of the eigenfaces. We detect a face by comparing the distance of the test image from the Facebook vectors; if the test image is too far away from all of the eigenfaces, it is considered to not have a face. We recognize a face by finding the Facebook vector closest to the test image.

5.5 Tracking

Tracking models the movement of an *object of interest*, also known as a *target*, over a sequence of observations. Observations can be made using sensing methods other than video, with radar tracking being a prime example. In this section, we will concentrate on a connected set of observations from a single sensor. We first consider the separation of objects of interest from background items. Section 5.5.2 looks at tracking from a single camera. Section 5.5.3 develops appearance models. Section 5.5.4 studies algorithms for activity analysis. Section 5.5.5 looks at tracking from a moving camera.

5.5.1 Background Elimination

The term *background elimination* (or sometimes *background subtraction*) is used to describe the process of identifying a subset of pixels for further analysis; typically, subjects that display more motion are identified as *foreground* and areas with less motion as background. Unfortunately, the use of the term *background* in this case

does not conform to standard usage. In theater, for example, *foreground* means the front of the stage and *background* means the back of the stage. We will use the computer vision terminology here to be consistent with the literature, but we do so under protest.

A naïve approach to background elimination is to take a reference frame known to not have any regions of interest and then to compare each successive frame to the reference frame. The two frames are compared pixel by pixel against a threshold:

$$\mathrm{BG}(i,j) = R(i,j) - I(i,j) < t. \tag{5.20}$$

The result is a background map that identifies each pixel as being either background or foreground.

This method is fast bit gives poor results in all but the most controlled situations. The typical scene includes small object movements that are not of interest. Figure 5.4 shows a light rail station with several objects that move or change subtly: the electronic sign changes and its scanning logic can create problems for image capture; the trees may move in the wind. Even indoors, small movements such as the placement of coffee cups may not be of interest.

The *mixture-of-Gaussians* methodology [Sta99, Sta00] provides a more robust method to separate small motions from the region of interest. This approach not only models pixel values as Gaussian, but it keeps several models for each pixel. The different models for a pixel may cover, for example, the case in which a leaf is visible at the pixel and when it is not. Typically, $K = 4$ models are kept. The pixels are modeled as independent. In order to reduce computational expense, Stauffer and Grimson also assumed that the color channels were independent and had equal variances. The probability of observing a pixel X_t is

$$\Pr(X_t, \mu, \Sigma) = \sum_{1 \le i \le K} \omega_{i,t} \eta(X_t, \mu, \Sigma) \tag{5.21}$$

where $\eta()$ is the Gaussian probability density function. The weighting factor $\omega_{i,t}$ is both position and time dependent. At each update, the weights are adjusted by

$$\omega_{i,t} = (1 - \alpha)\omega_{i,t-1} + \alpha(M_{k,t}) \tag{5.22}$$

where the decision variable $M_{k,t} = 1$ if the model was matched and 0 otherwise. The weights are renormalized after all the pixels have been reweighted. The mean and variance for a matched distribution are updated as

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{5.23}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{5.24}$$

where

Fig. 5.4 A scene with small amounts of movement

$$\rho = \alpha\eta(X_t, |\mu_k, \sigma_k). \tag{5.25}$$

Schlessman et al. [Sch07] designed a hardware implementation of the mixture-of-Gaussians approach using three major functional units: comparison, updating of means and variances, and updating of Gaussian weights. They identify as the background models those with the least variance and which are most widely represented in the image. They use T as an estimate of the proportion of the image that should be considered background. They choose as background models the first B distributions such that

$$B = \text{argmin}_b \left[\sum_{1 \le k \le b} \omega_k > T \right]. \tag{5.26}$$

Sheikh and Shah [She05] used competing background and foreground models to improve target/background separation. They assumed that targets are relatively constant in their appearance. They used a Gaussian model over a range of pixels to take into account correlations between pixels. An edge-preserving Markov random field estimates target position.

5.5.2 Tracking from a Fixed Camera

The simplest view of tracking is as a historical problem—we identify the target in each frame and record its position. However, we typically treat tracking as a

prediction problem—based on the target's recent behavior, its position in the next frames is predicted.

The basic model of motion is linear with unchanging direction. We can use a *Kalman filter* to track the target. For simplicity, we will formulate the target position in image coordinates, which can be separately translated to world coordinates. The state of the target in frame k includes both its position and velocity:

$$\boldsymbol{x}_k = \begin{bmatrix} x_k & y_k & v_{xk} & v_{yk} \end{bmatrix}. \tag{5.27}$$

An observation in frame k only indicates its position:

$$\boldsymbol{y}_k = \begin{bmatrix} y_x & y_y \end{bmatrix}. \tag{5.28}$$

The system state and our observation of that state is updated from frame to frame as

$$\boldsymbol{x}_k = \Phi \boldsymbol{x}_{k-1} + \xi, \tag{5.29}$$
$$\boldsymbol{y}_k = H\boldsymbol{x}_k + \mu. \tag{5.30}$$

The state transition matrix Φ updates the position based on the velocity components:

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5.31}$$

The measurement matrix H extracts the position from the state vector. ξ represents model uncertainty, while μ represents observation noise. We write the prediction as $\widehat{\boldsymbol{x}_k}$ or, when computed before the latest observation, $\widehat{\boldsymbol{x}_k}^-$. Let the covariance matrix of the state be P and the covariance of the estimate be R. We estimate the new system state in two steps. The *propagate* step finds

$$\widehat{\boldsymbol{x}_{k-1}}^- = \Phi \widehat{\boldsymbol{x}_k}, \tag{5.32}$$
$$P_{\boldsymbol{k}+1}^- = \Phi P_k \Phi^T + Q. \tag{5.33}$$

The *update* step involves both the position estimate and the covariance matrix:

$$P_k = \left(\left(P_k^- \right)^{-1} + H^T R^{-1} H \right)^{-1}, \tag{5.34}$$
$$\widehat{x_k} = \widehat{\boldsymbol{x}_k}^- + P_k H^T R^{-1} \left(\boldsymbol{y}_k - H\widehat{\boldsymbol{x}_k}^- \right). \tag{5.35}$$

We can also formulate tracking as a hidden Markov model. The state of both the subject $p(x_k | x_{k-1})$ and our observation of the subject $p(z_k | z_{k-1})$ depends on those values from the previous frame. The probability of an extended track and our observations of it in the interval $[1, \cdots, k]$ can be written as

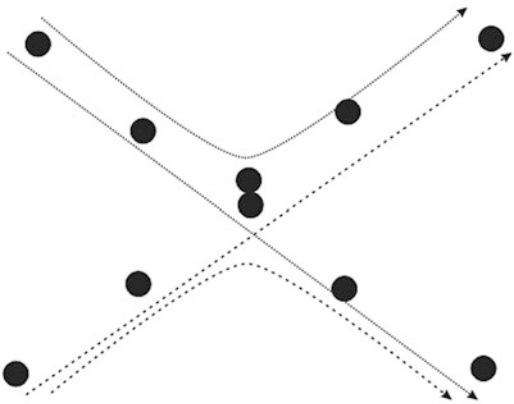


Fig. 5.5 Possible paths from observations of multiple targets

$$p(x_1)p(z_1|x_1)\prod_{2\le i\le k}\{p(x_i|x_{i-1})p(z_i|x_i)\}. \tag{5.36}$$

We will return to this type of model in Sect. 5.6.3.

If more than one target is being tracked, the tracker needs to be able to assign observations to distinct tracks. A classic challenge case for multitarget tracking is shown in Fig. 5.5. The two targets are initially far apart but converge to cross a common point at about the same time. The observations can be assigned to two radically different track assignments: each target continues along its initial trajectory after the crossing, or each target changes its course to follow a reflection of the earlier path.

Reid [Rei79] used a Kalman filter to track each target; he assumed that the number of targets was known a priori. He built hypotheses trees to keep track of the possible target assignments for each observation. He used a Bayesian model to model a hypothesis derived from a set of observations. P_i^k is the probability of a given configuration at time k using the observations over the period $[1,\cdots,i]$. N_{DT}, P_{DT} are the number and probability of detection; N_{TGT} is the number of previously known targets. $N_{\mathrm{FT}},\beta_{\mathrm{FT}}$ are the number and probability of false targets, while $N_{\mathrm{NT}},\beta_{\mathrm{NT}}$are the number and probability of new targets. The probability of a new configuration of targets is

$$P_i^k = \frac{1}{c}P_D^{N_{DT}}(1-P_D)^{(N_{TGT}-N_{DT})}\beta_{FT}^{N_{FT}}\beta_{NT}^{N_{NT}}\left[\prod_{1\le m\le N_{DT}}N\left(y_m - H\bar{x}, P_{\boldsymbol{k}}^-\right)\right]P_g^{k-1} \tag{5.37}$$

In this formula, c is a normalization constant.

Fortmann et al. [For83] formulated the multitarget tracking problem by building hypotheses to assign observations to targets. Each observation should be associated with no more than one event. Let $\tilde{y}_j = y_j - \hat{y}$ and S be its covariance. The probability density of a measurement y_i association with a target t is Gaussian:

$$N\left(\tilde{y}_j^t; 0, S^t\right) = \frac{\exp\left(\left(-\tilde{y}_j^{t\prime} S_t^{(-1)} \tilde{y}_j^t\right)/2\right)}{(2\pi)^{(M2)} |S_t|^{1/2}}. \tag{5.38}$$

The probability of a set of target identifications given a set of observations is

$$P(X|Y_k) = \frac{C}{c} \prod_{j \epsilon \tau} N\left(\tilde{y}_j^t; 0, S^t\right) \prod_{t \epsilon \delta} P_D^t \prod_{t \epsilon \neg \delta} \left(1 - P_D^t\right). \tag{5.39}$$

C is in this case the density of false measurements, τ is the set of measurements associated with a valid target, and δ is the set of detected targets.

5.5.3 Appearance Models

Several types of problems can present themselves in imagery for the tracking problem:

- The target may be *occluded* by other objects. In the case of multiple targets, one target may occlude another.
- The lighting on the target may change.

Mixed indoor-outdoor lighting presents the greatest challenges for any sort of image interpretation. These photos often exhibit very wide dynamic range that may be beyond the capabilities of the image sensor. Lighting conditions can also change nearly instantaneously, for example, as clouds move onto the scene. Lighting conditions also change slowly over the course of the day so that settings which work at noon no longer work at night. Figure 5.6 shows an example of a train platform: during the day, the track is brightly lit and the platform is in shade; at night, the platform is more brightly lit than is the track.

We need an *appearance model* for the target and a criterion for the similarity of two appearance models. Background subtraction identifies a set of pixels considered to be foreground. A simple shape model for the foreground object is its bounding box. Given a set of pixels $p_i \epsilon P$, its bounding box $\langle LL, UR \rangle$ is

$$\langle \langle \min x(P), \min y(P) \rangle, \langle \max x(P), \max y(P) \rangle \rangle. \tag{5.40}$$

Since the size of the target will change depending on its distance from the camera, we need relative metrics for comparison of two bounding boxes. The simplest metric is the ratio of the bounding boxes B_1, B_2:

$$\frac{|UR_{1x} - LL_{1x}| / |UR_{1y} - LL_{1y}|}{|UR_{2x} - LL_{2x}| / |UR_{2y} - LL_{2y}|}. \tag{5.41}$$

Fig. 5.6 Lighting on a train platform

The histogram of the bounding box region can be used to further characterize its appearance; either the color or luminance histograms can be used. The *Bhattacharyya distance* is commonly used to compare histograms and other distributions. The Bhattacharyya distance between histograms H_1, H_2 is

$$-\ln \sum_i H_1(i)H_2(i) \tag{5.42}$$

where i ranges over the histogram bins.

Jepson et al. [Jep03] used a multicomponent appearance model. An observation is d_t.The subject appearance is assumed to have a stable set of features described as a Gaussian q_t. It models data outliers as uniformly distributed and denoted as $P(d_t)$. The third component of the model has a short time constant to account for either motion or sudden changes in appearance (people, e.g., change appearance when they turn to face away from the camera). The probability of an observation corresponding to the subject is given by the mixture

$$P(d_t|\boldsymbol{q_t}, \boldsymbol{m_t}, d_{t-1}) = m_w P(d_t|d_{t-1}) + m_s P(d_t|\boldsymbol{q_t}) + m_l P(d_t) \tag{5.43}$$

where $\boldsymbol{m} = \{m_w, m_s, m_l\}$ are the mixing probabilities. The main parameters—the Gaussian parameters for the stable component and the mixing probabilities—are

estimated using *expectation maximization*. The log-likelihood of the observation history is formulated using a support envelope $S_t(k)$:

$$L(\boldsymbol{d}_t|\boldsymbol{q}_t, \boldsymbol{m}_t) = \sum_{t \le k \le -\infty} S_t(k) \log P(d_k|\boldsymbol{q}_t, \boldsymbol{m}_t, d_{k-1}). \tag{5.44}$$

The expectation step computes the ownership probabilities for the observations:

$$o_{i,t}(d_k) = \frac{m_{i,t} P(d_k; \boldsymbol{q}_t, d_{k-1})}{P(d_k; m_t, \boldsymbol{q}_t, d_{k-1})}. \tag{5.45}$$

The maximization step updates the mixture probabilities and the Gaussian mean and variance of the stable component:

$$m_{i,t} = \sum_{t \le k \le -\infty} S_t(k) o_{i,t}(d_k), \tag{5.46}$$

$$\mu_{s,t} = \frac{M_{1,t}}{M_{o,t}}, \tag{5.47}$$

$$\sigma^2_{s,t} = \frac{M_{2,t}}{M_{0,t}} - \mu^2_{s,t}. \tag{5.48}$$

The M_s are moments defined by

$$M_{j,t} = \sum_{t \le k \le -\infty} S_t(k) d_k^j o_{i,t}(d_k). \tag{5.49}$$

To reduce storage requirements, they approximate the current ownership as the ownership at the time for which the data was first observed.

Comaniciu et al. [Com03] used a *mean-shift* target model. They used a generalized form of the Bhattacharyya coefficient to compare target models; they denoted this similarity function as $\hat{p}(y) = \rho[\hat{p}(y), \hat{q}]$. They represent the target as an ellipse with normalized size. A *kernel profile* $k(x)$ weights pixels relative to their distance from the center; the kernel profile is convex and monotonic decreasing. For the example of the Bhattacharyya coefficient as a similarity function, the distance between two candidate distributions $\hat{p}, \hat{q}$ depends on a set of weights:

$$w_i = \sum_{1 \le u \le m} \sqrt{\frac{\widehat{q_u}}{\widehat{p_u(y_0)}}} \delta[b(x_i) - u]. \tag{5.50}$$

In this formula, x is the position of the target in the original and y is its position in the new frame, while the function $\delta[b(x_i) - u] = 1$ when u is equal to the bin holding pixel x_i. A new estimate of the kernel position, moving from position $\widehat{y_0}$ in one frame to position $\widehat{y_1}$ in the next frame, is given by

$$\widehat{y_1} = \frac{\sum\limits_{1 \le i \le n_h} x_i w_i g\left\| \frac{\widehat{y_0} - x_i}{h} \right\|^2}{\sum\limits_{1 \le i \le n_h} w_i g\left\| \frac{\widehat{y_0} - x_i}{h} \right\|^2} \tag{5.51}$$

This formula is used to iteratively estimate the new position. At each step, the similarity function is tested and compared to a threshold, typically chosen to give pixel-level accuracy for the search. If the similarity function is larger than the threshold, the position is adjusted as $\widehat{y_1} \leftarrow \frac{1}{2}\left(\widehat{y_0} + \widehat{y_1}\right)$.

Zhou et al. [Zho04] used a mixture-of-Gaussians model to take into account variations in appearance of the target, such as the target turning to present a different aspect of its features to the camera. The observation likelihood is a product of Gaussians. The target is modeled directly using pixel values. As a result, their update process is similar to that used for background elimination.

5.5.4 Activity Analysis

Activity analysis can take into account combinations of people moving through spaces and the gestures and poses they make.

Pfinder [Wre97] was an early real-time human activity tracker. Its algorithms were based on statistical models of blobs that represent identified portions of the person. The system did not explicitly use background elimination. It did rely on initial capture of a scene without a subject and allowed slow changes in the background. The subject was then expected to enter the scene; contour analysis was used to build an initial set of blobs based on a simple model of the person with an arms-extended pose commonly used to build an initial set of blobs. Each blob matrix was represented as a Gaussian model with mean $\boldsymbol{\mu}$ and covariance $\mathbf{K}$. Each blob also had a support map with Boolean per-pixel entries indicating whether the blob occupied that pixel. In addition to the occupancy of the frame, each blob was represented by a YUV color vector to represent the overall color of the blob. Pixels not occupied by blobs are modeled as a YUV color value with mean $\boldsymbol{\mu}_0$ and covariance $\mathbf{K}_0$. At each frame, each pixel not occluded by a blob is updated using the rule $\boldsymbol{\mu}_t = \alpha y + (1 - \alpha)\boldsymbol{\mu}_{t-1}$. Each blob is updated using the Kalman-style rule

$$\widehat{\boldsymbol{X}_{[\boldsymbol{n}|\boldsymbol{n}]}} = \widehat{\boldsymbol{X}_{[\boldsymbol{n}|\boldsymbol{n}-1]}} + \widehat{\boldsymbol{G}_{[\boldsymbol{n}]}}\left\{\widehat{\boldsymbol{Y}_{[\boldsymbol{n}]}} - \widehat{\boldsymbol{X}_{[\boldsymbol{n}|\boldsymbol{n}-1]}}\right\} \tag{5.52}$$

where G is the Kalman state matrix of EQ 5 Kalman state. At each frame, given a pixel value $\boldsymbol{y} = [x\ y\ U/Y\ V/Y]$, they find the likelihood for each pixel's membership k in the set of blobs and background:

$$d_k = -\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu_k})^T \boldsymbol{K_k}^{-1}(\boldsymbol{y} - \boldsymbol{\mu_k}) - \frac{1}{2}\ln|\boldsymbol{K_k}| - \frac{m}{2}\ln 2\pi. \tag{5.53}$$

They use heuristics to deal with luminance changes caused by shadows. They use these likelihoods to build support maps for each of the blobs and for the background. They use the new member pixels of a blob to update the model mean and covariance:

$$\left(\widehat{\boldsymbol{\mu}}_k\right) = E\left[(\boldsymbol{y} - \boldsymbol{\mu}_k)(\boldsymbol{y} - \boldsymbol{\mu}_k)^T\right], \tag{5.54}$$

$$\left(\hat{\boldsymbol{K}}_k\right) = E\left[\boldsymbol{y}\boldsymbol{y}^T\right] - \boldsymbol{\mu_k}\boldsymbol{\mu_k^T}. \tag{5.55}$$

Wolf et al. [Wol02] analyzed gestures in real time. Their algorithm did not use markers but was intended to classify gestures, not track the motion of limbs. They used background elimination to separate the subject from the background and then identified boundaries for parts of the foreground. They fit an ellipse to each of the region, which was then modeled as its ellipse parameters plus a bit for flesh tone/non-flesh tone color. Figure 5.7 shows the region boundaries and the ellipses fitted to those regions; the ellipses are also labeled by color, either flesh tone or non-flesh tone. A graph was built with a node for each ellipse and edges between adjacent region nodes. The graph was matched against a library of poses to classify the pose of that frame. A hidden Markov model was then used to classify a sequence of poses into a gesture.

Stauffer and Grimson [Sta00] developed a codebook-based classification algorithm for the classification of motion. An observation is of the form [*x y dx dy size*]. They used online vector quantization to generate a codebook of a set of observations. The activity of a target does not directly model time or sequence; an activity is a multiset (which may contain multiple instances of a given element) of observations. They build a co-occurrence matrix for pairs of prototypes i, j such that $c_{i,j}$ represents the probability of an observation corresponding to the i^{th} prototype being followed by an observation represented by the j^{th} prototype. They use this co-occurrence matrix to build probability mass functions (PMFs) for sequences. They build the PMFs in the form of a binary tree with N nodes. The co-occurrence matrix can be estimated by the PMFs $p_i()$ and the prior probabilities π_i for the sequences:

$$\widehat{C_{i,j}} = \sum_{1 \le c \le N} \pi_c{}^* p_c(i)^* p_c(j). \tag{5.56}$$

The priors and PDFs are iteratively estimated to minimize the sum-squared error of the co-occurrence matrix estimate. The updated rules are

$$\pi_c \leftarrow (1 - \alpha_\pi)^* \pi_c + \alpha_\pi \sum_{i,j} \left(C_{i,j} - \widehat{C_{i,j}}\right) p_c(i)^* p_c(j), \tag{5.57}$$

Fig. 5.7 Ellipse models for gesture analysis

$$p_c(i) \leftarrow \left(1-\alpha_p\right)^* p_{c(i)} + \alpha_p \sum_{i,j} \left(C_{i,j} - \widehat{C_{i,j}}\right)^* p_c(j). \tag{5.58}$$

They used $\alpha_\pi > \alpha_p$. At each branch in the modeling tree, the branch l co-occurrence matrix is derived from the parent co-occurrence matrix as $C^l_{i,j} = C_{i,j}{}^* p_l(i)^* p_l(j)$.

Brand and Kettnaker [Bra00] used hidden Markov model training based on entropy minimization to improve results on small video datasets.

5.5.5 *Tracking from a Moving Camera*

Tracking from a camera on a moving platform—such as a car—requires a great deal of analysis to be able to perform the tracking itself. *Egomotion* is the motion of the camera relative to the scene. In order to track the target relative to the scene, we need to determine and subtract out egomotion. We must do so without prior knowledge of the camera parameters and, in the face of complex, noisy movement. The video stabilization methods of Section 4.stab were intended for aesthetic use and did not require extreme accuracy. Egomotion analysis for tracking from a moving platform requires substantially more accuracy.

Tracking from a moving camera is typically performed in several steps [Yam06]:

- Feature points are extracted for each frame.

- Corresponding features are identified between frames i and $i+1$.
- Egomotion is estimated from the corresponding features.
- The 3D structure of the scene is estimated and a region of interest is identified. For example, in the case of vehicles on roads, the lane is recognized; a region some distance ahead may be the region of interest.
- The target is identified.

Even though we are dealing with a single camera, we need to choose a projection of the scene in order to at least approximately correct for perspective effects. Kang et al. [Kan05] performed an initial affine motion detection and used the results to perform an additional step to minimize parallax. Lin [Lin12] used an ellipsoid model for the surface in front of the vehicle to approximate the perspective effects in a typical driving scenario. A motion vector in the scene is subjected to a translation T and rotations ω to project it onto the image surface. If distances to objects in the scene are large compared to the focal length, we can assume small rotation angles. This ellipsoidal model gives ten degrees of freedom: focal length, three rotational, three translational, and the three ellipsoidal parameters.

We need feature points for both egomotion analysis and target tracking. We can use a variety of methods to generate features, such as Harris corners or SIFT/SURF. Figure 5.8 gives an example of egomotion point matching: red crosses are features from frame $i-1$, green crosses are from frame i, and blue points are estimations.

We estimate the egomotion vector by comparing the identified motion vectors (red to green in the figure) to our estimate of the motion and the resulting motion vector (red to blue). This objective function can be written as

$$\sum \left\| \vec{g_i^t} - \vec{h_i^t} \right\|. \tag{5.59}$$

This objective function can be minimized by making an initial guess for the value of the motion, using genetic algorithms to construct an improved solution, and then using nonlinear least squares optimization to find the final value.

We can use the egomotion result to perform background elimination; the purely pixel-oriented algorithms of Sect. 5.5.1 are clearly insufficient when the camera moves with every frame. We can use the egomotion vector to project the previous frame onto the current frame; we can interpolate pixel values to improve accuracy. An example is shown in Fig. 5.9. While edges of objects in the scene introduced a small amount of noise, the largest motion corresponds to objects that moved: the car, a pedestrian, and leaves on a tree. We can use Bayes' rule to classify pixels as background or foreground (along with the fact that $P(fg)+P(bg)=1$):

$$P(b|v_k, s, \text{mp}) = \frac{P(v_k|b, s, \text{mp})P(b|s, \text{mp})}{P(v_k|s, \text{mp})} \tag{5.60}$$

where mp is the camera motion parameter set, s is the pixel position, and v_k is the pixel value. We can estimate these probabilities using histograms:

Fig. 5.8 Matching features between frames [Lin12]

Fig. 5.9 An example comparison between a current frame and a compensated previous frame [Lin12]

$$P(b|v_k, s, \text{mp}) \approx \frac{H_{vb}^t P_b^t}{H_v^t}. \tag{5.61}$$

We can estimate $P_b^t(s)$ using the interpolated version of the previous frame.

We can group together motion vectors using clustering algorithms. The number of clusters varies from frame to frame so we need to use bottom-up clustering algorithms to identify groups of similar motion vectors. We can estimate *linkages* between clusters in the previous and current frame [Lin10]; some clusters in the previous frame will link to only one cluster in the current frame, while others may

have several possible matches. The result is a directed graph whose nodes are the clusters for each frame and with time-oriented directed edges from one linked cluster to the next. This model is an example of a *Bayesian network*.

We can use *belief propagation* to classify motion [Lin10]. This class of algorithms uses graph traversal to find the marginal distributions of hidden nodes based upon the observed nodes. In the case of motion vector clustering, the relevant observed variables include the average angle of motion vectors in a group, the number of feature points in the group, and the distance between groups.

5.6 Multicamera Systems

Multiple cameras provide us with information about a scene that we cannot obtain from a single camera. Training multiple cameras on a scene helps us with four different problems:

- *Occlusion* is caused from other objects (a person standing behind a desk) or by the object itself (front view vs. back view). Cameras at different positions help us to cover more of the objects in the scene.
- *Pixels on target* is a useful metric for resolution. As a subject moves away from a camera, the subject's image falls on fewer and fewer pixels. When another camera is placed at the right position, it will see the subject move toward it.
- *Depth* can be recovered from disparity.
- *Larger spaces*—buildings, parks, and cities—can be covered by multiple cameras in ways that cannot be achieved by a single camera.

Multicamera systems are used in many applications. Motion capture for video games uses several tightly coordinated cameras in a structured environment. Surveillance systems make use of multiple cameras that are usually less tightly coupled: Tokyo Station makes use of over 700 cameras in a relatively small space; the cities of London and Chicago combine cameras operated by the city with information from privately owned cameras to provide assistance to law enforcement.

Multicamera systems introduce purely algorithmic questions but, as we will see in the next section, they also require consideration as distributed computing systems. After looking at the need for distributed algorithms, we will study calibration algorithms in Section 5.multicalibration and tracking algorithms in Section 5.multitracking.

5.6.1 Multicamera Systems as Distributed Computing Systems

Early multicamera systems sent video signals to a centralized computer for processing. This approach simplified many problems that allowed progress on algorithms. But centralized processing does not scale for video analysis, just as with many other applications. *Distributed algorithms* running on *distributed computing systems* are required to provide real-time video analysis.

The first limitation of centralized video analysis is *latency*. As we will see in Section 5.multicalibration, comparing video streams requires synchronizing them. Early multicamera systems used analog cameras; these cameras had to be run off a common clock; the video cable lengths had to carefully controlled to maintain synchronization. As a result, only systems with very small diameters could be built. Digital video streams eliminate some of these possibilities, but longer network distances still introduce delay. Buffering can be used to maintain logical synchronization, but as a result, the latency from image capture to analytical result increases. Distributed algorithms reduce the radius of communication, thus reducing both latency and buffer memory requirements.

Distributed algorithms also benefit video applications for the same reasons as with other applications: reduced bandwidth, lower communication power consumption, and lower computational power consumption. Bandwidth and power consumption are particularly important for multicamera systems because they are inherently physically distributed *Internet of things* (*IoT*) systems. Installation cost is a critical component of cost of ownership of IoT systems, and this cost is dominated by the cost of pulling wires to the camera's location. Wireless networks offer reduced installation cost, but their bandwidth is limited relative to wired networks. Power wiring is also a critical cost, so reducing power consumption is key to installation and maintainability; communication power consumption is a critical component of overall power consumption.

Distributed computing also offers increased privacy. Given the powerful abilities of modern embedded computing platforms, we can process raw video within the camera and transmit only abstract representations. Assuring users that no raw imagery leaves the camera ameliorates the privacy concerns of many people. (We should note that in many instances, people are even more concerned about audio privacy than they are about their visual privacy.) Widen [Wid08] analyzes the law relating to privacy and video surveillance.

We need to carefully consider the abstractions that the cameras present to each other. How we represent imagery as something other than pixel arrays determines the way we combine information from multiple cameras. Abstract representations also influence power consumption, bandwidth, and latency.

Consider the example of Fig. 5.10. The cameras have overlapping fields-of-view, but each camera can see parts of the scene that the other cannot. As the subject moves, it starts in the field-of-view of camera A, then to the overlapping views of A and B, and finally to a position at which it can be seen only by B. It

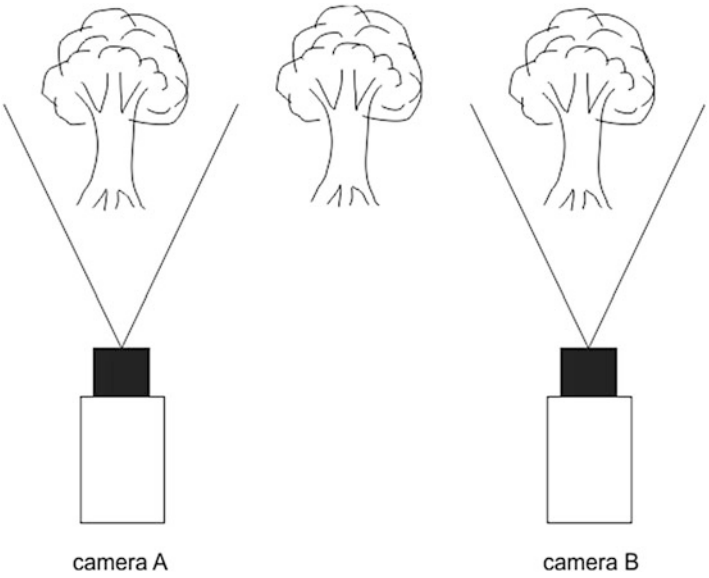


Fig. 5.10 Moving the computation with the subject

makes sense to move the computation as the subject moves; so long as one camera handles the entire analysis, all we need is a simple handoff mechanism. However, the situation can be more complicated in the overlap region. The subject may be located at a point at which neither camera has a full view of the subject. They can cooperate at different levels of abstraction: one can send its pixels to the other or they can trade models at some level of abstraction.

Lin et al. [Lin10B] designed a distributed version of the gesture recognition system. When only part of the subject is in view of each camera, each camera performs low-level computations; one of the cameras is designated to perform the data fusion of these operations to perform the final gesture classification. A token is passed around the network to identify the lead camera node for the subject; the token management protocol was formally verified. They point out that the amount of data that must be shared between nodes varies widely depending on the level of abstraction: entire standard definition images require 100 kB; contour points require 2–5 kB; and ellipse parameters or body part parameters each require fewer than 100 bytes. If the cameras trade ellipse parameters, they must determine whether the ellipse crosses camera boundaries; in this case, they must share lower-level data to build an accurate model of the body part. The system was built on a service-oriented model; middleware provided node management, service discovery, and service scheduling.

Distributed algorithms are harder to design than centralized algorithms. Video algorithms add the challenge of temporal synchronization, but we must also consider algorithmic *synchronization* to ensure that each thread of the computation has the data it needs. We need *agreement algorithms*, a simple example being

identifying a common label for an object that can be seen by several cameras. Distributed computing models result in more complex programming models and APIs.

Given the difficulty of distributed program design, we want to isolate the application designer from details of application development. We can use *middleware* [Rin08] to provide abstractions to the application developer that hide details of operating system mechanisms and provide higher-level interfaces to support distributed computing and the particular complexities of distributed smart cameras. An *agent-based model* allows a computation to migrate from node to node. *Agent models* require data migration and sometimes code migration; they are particularly challenging for heterogeneous platforms in which different binaries are required to run on different components of the platform. A *publish/subscribe* system provides message-based communication without requiring the publisher of a message to explicitly concern itself with the identity of nodes which will receive the message. *Quality-of-service* (*QoS*) managers allow applications to specify bandwidth requirements and then manage the priority of communications to ensure that each application receives the bandwidth it was promised.

5.6.2 Multicamera Calibration

Spatial calibration in cameras with overlapping fields-of-view requires finding points in the scene which can be used for correspondence. Figure 5.11 shows a scene viewed from two different positions. The field-of-view line of the other camera is marked in each image. Three reference points on the ground that are visible to both cameras are also marked; these three points are sufficient to define the ground plane.

If we have no information about the location of any point in the scene, cameras can be calibrated up to a similarity transformation [Har03] Devarajan et al. [Dev06] calibrate a camera network using a distributed algorithm. The camera model is based on perspective:

$$P_i = K_i R_i^T \left[I_{(3\times 3)} - C_i \right]. \tag{5.62}$$

K_i is the camera's intrinsic matrix, R_i is the camera rotation matrix, and C_i is the camera center.

We refer to the set of points in the scene used by the camera network for calibration as $\{X_1, \cdots, X_N\}$. The projection of one of these points X_j by camera i is

$$\lambda_{ij} \begin{bmatrix} u_{ij} \\ 1 \end{bmatrix} = P_i \begin{bmatrix} X_j \\ 1 \end{bmatrix}. \tag{5.63}$$

λ_{ij} is the projective depth of the point at the camera.

Fig. 5.11 Corresponding points for spatial calibration

They model the camera system using two graphs. The *vision graph* with M nodes describes pairs of cameras which share views of a certain minimum number of calibration points. The *communication graph* describes camera pairs that can directly communicate with each other.

They form an initial vision graph by first identifying feature points in each camera; they used both corner detection and SIFT to identify features. They then identify initial nearest-neighbor matches between features in image pairs and then perform outlier rejection using both extremal heuristics and a RANSAC-style operation.

Each camera locally calibrates itself using a version of structure from motion—the camera estimates the positions of the calibration points and camera parameters using 2D image correspondences. The set of image projections has the form

$$W = \begin{bmatrix} P_1 \\ P_2 \\ \vdots \end{bmatrix} \left[X_1^h X_2^h \cdots \right] \tag{5.64}$$

Solving for the projective depth values runs into ambiguities which can be resolved using the absolute dual quadric matrix [Har03]. Based on this initial estimate, *bundle adjustment* is used to improve the solution. This optimization procedure minimizes the cost function

$$\min_{\substack{\left\{P_j^i\right\}, j \in \{i, C_i\} \\ \left\{P_j^i\right\}, k \in V_j}} \sum_j \sum_k \left(\widehat{u_{jk}} - u_{jk}\right)^T \sum_{jk} \left(\widehat{u_{jk}} - u_{jk}\right)^{-1} \tag{5.65}$$

The resulting parameters are expressed relative to its neighbors. They form clusters of cameras and associated points to translate global positions.

Cameras must also be calibrated temporally. Errors in time between cameras correspond to errors in tracking position. Cameras exhibit wide variations in frame rate; even if they are synchronized at one point in time, their frame rates are not accurate enough to ensure a consistent time base. Velipasalar and Wolf [Vel08] used tracking to temporally calibrate distributed cameras. Two cameras are calibrated using a tracking target that can be seen by both cameras. Correspondences between locations in the two images are found using four pairs of points on the ground plane of the scene. Given these correspondences, the position of the target as projected onto the ground plane can be found using the projective invariant formulas. After each camera generates its own track for the target, they compare their positions for the target. Local search is used to find the correspondence between frames such that both cameras register the same position for the target at the same time. This algorithm is fast enough to be used periodically to keep the cameras synchronized.

5.6.3 Multicamera Tracking

The most direct generalization of tracking to multicamera systems is to assume that the fields-of-views of the cameras at least partially overlap. Continuous coverage is important in many surveillance applications.

We can build a distributed tracking system as a network of *cooperating trackers:* each camera performs its own tracking and the cameras also comparing their results with those of other cameras [Vel05]. The cameras can identify the relationships between their fields-of-view using multicamera calibration methods; it is also possible to find field-of-view lines without full calibration [Kha03]. Based on those results, each camera can determine the field-of-view lines for each of the other cameras with which it shares views of the scene. Cameras need to agree with each other as to the identity of the subjects being tracked. When a subject enters the field-of-view of camera i, that camera can determine the other cameras which can see the subject based on the field-of-view lines. It can then communicate with the other camera to compare the position and appearance of the target. If the two agree that these observations correspond to the same target, they can assign a common label to identify the target. If the target is later occluded from the camera's view, the camera can obtain the target's position from the other cameras. Figure 5.12 shows the results of tracking a model car from a set of three cameras placed 120° apart. The cameras have agreed on label 51 for the target. The box completely occludes the subject in the bottom view, but the camera is able to determine the subject's position from the other cameras. Foreknowledge of the occluded object's position helps the camera processes to be ready for the subject's reappearance. A protocol was used to control the cooperation of cameras [Vel06]. The protocol provided non-blocking communication between the cameras. A camera could send a

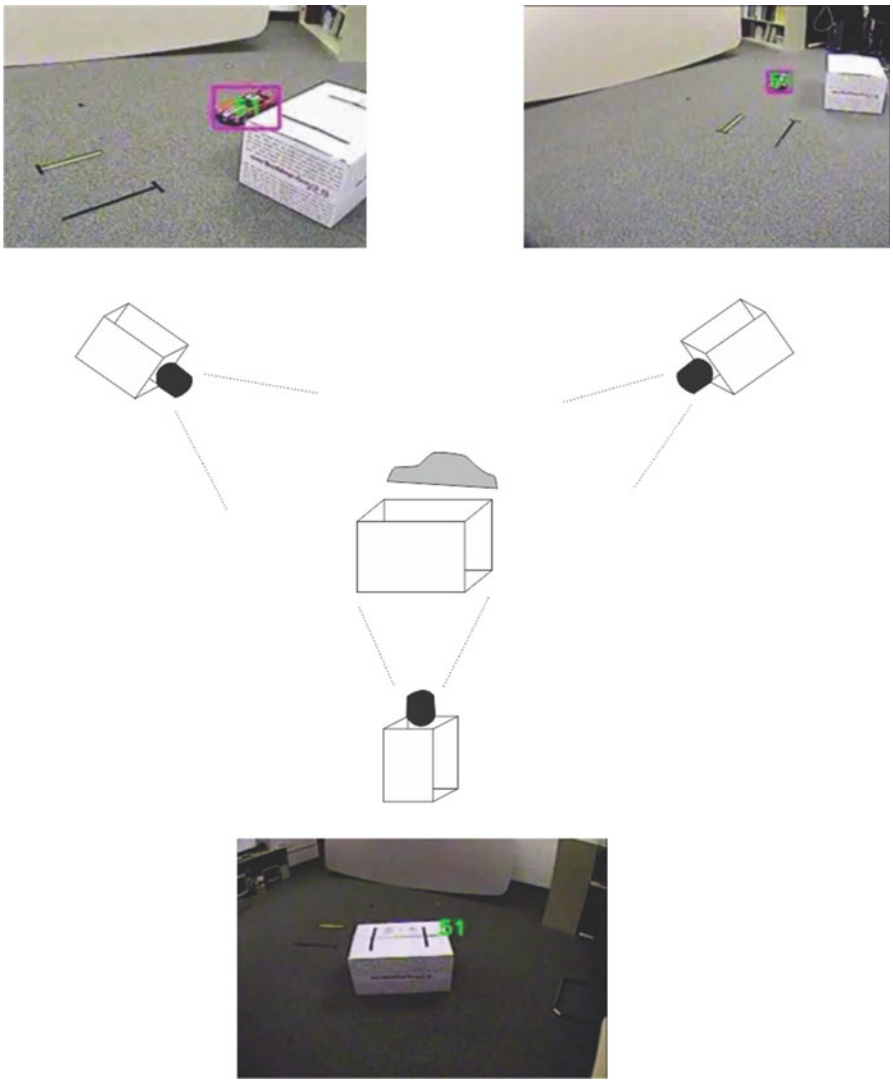


Fig. 5.12 Tracking in the presence of occlusion using cooperating cameras

message and then proceed with more processing without waiting for responses. Synchronization points could be used to ensure that all nodes had reached the same point; the cameras were synchronized every few frames. Tracking accuracy depends on the synchronization rate: synchronizing on every frame resulted in 95% accuracy, while synchronizing once every two seconds (60 frames) resulted in 55% accuracy. The MPI library was used to manage inter-camera communication.

Kokiopoulou and Frossard [Kok10] developed a distributed algorithm to classify targets into one of several possible classes based on the sensor readings from the cameras. A training phase generates a set of weights to capture the similarity of observations of a target from different cameras. Each node computes an initial smoothness function based on the trained weights. The nodes then exchange values in a consensus algorithm to agree on an assignment of the observations to a class.

However, overlapping fields-of-view are impractical for very large areas. Tracking from cameras with nonoverlapping fields-of-view is often formulated using techniques related to hidden Markov models.

Covering large areas ultimately requires placing cameras with nonoverlapping fields-of-view. This tracking problem combines traditional computer vision techniques with combinatorial optimization.

Chang and Gong [Cha01] used a Bayesian network to fuse tracking results from multiple cameras. Their network included four types of nodes: correspondence nodes encode assignments of observations to targets; comparison nodes compare a subject at one camera against possible subjects at another camera; modality confidence nodes represent the confidence of a given observation modality; and indicator nodes indicate modality confidence.

Javed et al. [Jav05] used a color calibration phase to correct for color rendering differences between cameras in the network. The training phase is used to generate brightness transfer functions between pairs of cameras in the network. They showed that these brightness transfer functions are of small dimension.

The tracking problem infers the activity of a set of subjects x^k from a series of observations. In the more general formulation, we have a series of observations; we do not know a priori the number of subjects or their appearance. Each camera produces observations $Y = \left\{\cdots, y_t^j, \cdots\right\}$ each at time t and with observation sequence number j; an observation includes an appearance model as well as a timestamp. We have an appearance model for each subject with a similarity metric $d(A(y_1), A(y_2))$. *Tracks* are the result of partitioning the observations into sets with each set $T_k = \{y_{t1}, \cdots, y_{tn}\}$ representing a track; the set of all tracks is $\omega_K = \{T_1, \cdots, T_K\}$. Each observation can belong to at most one path. We will refer to the actual behavior of the subject as a *path* ρ_i. The track is an approximation of the path, which is itself hidden.

In many applications, we can model the movement of targets as discrete paths. Hallways in buildings are a clear example, as are roadways. But even many open spaces such as parks may exhibit preferred pathways either as paved paths or as paths worn into the ground. Figure 5.13 shows an example *path graph* model for part of a building. A node in the graph represents a camera. The camera positions are arbitrary and need not be located at intersections. Directed edges connect two nodes $i \rightarrow j$ if the subject can move from i to j. We use a designated *entry*/*exit* node to model subjects entering or leaving the system. We can label the path graph with two probabilities: $P(v_j | v_i)$ is the *edge probability* of a subject moving from node i to j and $P(t | v_j, v_i)$ is the *travel time probability* for the time required to move from node i to j.

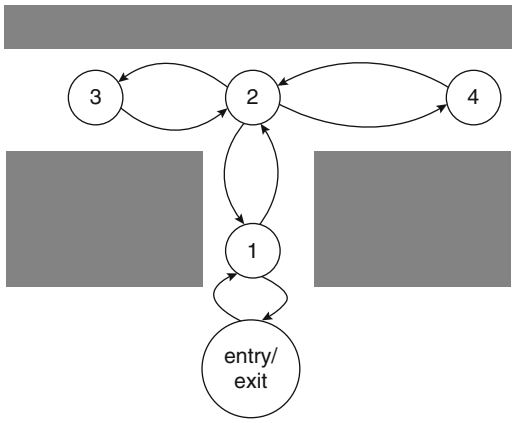


Fig. 5.13 A path graph

We can add directionality information to the observations to reduce uncertainty [Kim09]. The camera adds to the observation of the direction through which the subject moves through the camera's field-of-view. Each node in the vision graph becomes a supernode that is internally modeled as a small graph with nodes for the entry and exit points used by the camera.

The probability of a track depends on both the fit of the observations to the path and the likelihood of the path itself. A path that, for example, requires the subject to teleport from one side of the building to the other within one second has a low path probability—the travel time probabilities in the path graph do not make such an occurrence likely.

Kettnaker and Zabih [Ket99] decomposed paths into links between pairs of adjacent cameras; the posterior modeled the likelihood that observations at the two nodes were generated by the same target. They model a track with three components: probability $trans(c_{i,j-1}, c_{i,j})$ of the duration and location of each transit from one camera to another, the probability of a track of a given length, and the probability of new targets entering the system. They maximize the ratio of the posterior of a set of track assignments to the null hypothesis of each observation being in a separate track. After applying modeling assumptions, their objective function becomes

$$\frac{p(\omega_{li}|Y)}{p(\omega_{li}^0|Y)} \approx \prod_i \prod_{2 \le j \le li} \frac{p\left(y_{i,j,1}|y_{i,j-1}\right) p\left(trans\left(c_{i,j-1}, c_{i,j}\right)\right)(1-p_x)}{p\left(y_{i,j,1}\right) p_x \lambda_{loc(i,j)}}. \tag{5.66}$$

In this formula, p_x is the probability of a target exiting the system. After taking the log of the product formula, minimizing the sum can be formulated as a weighted assignment problem which can be solved using the Munkres algorithm.

We can write the probability of a path as the product of the link probabilities and of the probability of the subject entering the system:

$$P(\rho) = P(x_0) \prod_{1 \le i \le |\rho|} P(x_i | x_{i-1}). \tag{5.67}$$

The probability of a track given a set of observations is given by a similar chain of probabilities. For track k, the probability of two consecutive observations as the result of the hidden movement of the subject is $P(y_{k,i-1}, y_{k,i} | x_{k,i-1}, x_{k,i})$. Given the travel time probabilities and the appearance similarity metric, we can refine this probability as

$$P\left(y_{k,i}, y_{k,j} | x_{k,i}, x_{k,j}\right) = P\left(t_j - t_i | x_i, x_j\right) P\left(d(A(y_1), A(y_2)) | x_i, x_j\right). \tag{5.68}$$

Then the probability of a track is

$$P(T_k) = P\left(y_{k,0}, y_{k,1} | v_0, x_{k,1}\right) \prod_{1 \le i \le |T_k|} P\left(y_{k,i-1}, y_{k,i} | x_{k,i-1}, x_{k,i}\right). \tag{5.69}$$

The probability of a set of tracks given the observations is the product of the path and track probabilities:

$$P(\omega_K | Y) = \prod_{1 \le k \le K} P(\rho_k) \prod_{1 \le k \le K} P(T_k) \tag{5.70}$$

We need to assign both edge and travel time probabilities to the path graph. In the absence of a priori information about where people go and how long it takes them to get there, assigning paths as equally likely is often a good starting point. Travel times can be estimated using assumptions about velocity. These probabilities can be updated based on observations as the system operates. This formulation can be modeled as a bipartite graph [Kim10] with one set of nodes representing the observations and the other set representing the tracks; weighted edges give the assignment of the observations to tracks and the associated *posterior* probability.

Oh et al. [Oh04] used Markov chain Monte Carlo algorithms to solve the observation-to-track assignment problem. Starting with each observation in its own track partition, they probabilistically modify the partition until a termination criterion is met. They used several types of moves:

- *Birth* generated a new track from a set of singleton observations; *death* decomposed a track into its individual observations.
- *Splitting* broke a track into two pieces while *merging* combined two tracks into one.
- *Extension* added several observations to a track while *reduction* removed several observations.
- *Update* added a single observation to a track.
- *Switch* swapped sections of two different tracks.

At each step, they first randomly select a move and the track/tracks to which to apply the move to generate a new set of tracks ω'. They provisionally apply the move and then accept the move with probability

$$A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')q(\omega', \omega)}{\pi(\omega)q(\omega, \omega')}\right). \tag{5.71}$$

Zajdel and Kröse [Zaj05] used a form of multiple hypothesis tracking to solve the track partitioning problem, resulting in a reduced-complexity algorithm. They form an initial set of tracks based on a subset of the observations processed by an expectation-maximization algorithm. Their algorithm iteratively processes the remaining observations. Observations are added to the track set one at a time; each new observation is added to several different candidate tracks which are then evaluated and pruned.

The tracking problem can also be solved using distributed algorithms based on MCMC moves [Kim10]. Successive observations are likely to come from cameras that are nearby in the vision graph. As a result, the search for successors and predecessors is primarily local, allowing us to partition the search algorithm across the distributed system. The system is mapped onto a set of overlapping neighborhoods, based on a radius r of communication between cameras. Cameras can share their own observations with their neighbors. Each camera formulates its own estimates of paths related to its observations. Cameras in the neighborhood can vote on updated tracks to create a local consensus. In the event of a tie, the camera that generated the observations wins. The accuracy of the resulting tracks depends on the radius of communication, but experiments show that a very small radius of $r = 2$ works well.

Person reidentification is used to determine whether a person sighted at different times by nodes in a distributed camera network is, in fact, the same person. Gheissari et al. [Ghe06] broke images of a person into parts and generated signatures that are invariant to illumination and pose as well as clothing movement. Features are identified using a hue/saturation histogram and a set of edges. A series of frames is analyzed to identify a stable set of edges—for example, edges that represent the boundaries between garments rather than edges created by draping of the fabric. They group together sets of edges that have low cost in space and time. A greedy algorithm partitions the primitive regions into clusters. They use the Hessian affine invariant operator to generate a large set of points of interest and generate correspondences; this approach uses the large number of feature points generated to compensate for the instability of the sets. They use a model to generate correspondences to body parts (head, torso, arms, legs). Zheng et al. [Zhe13] use learning algorithms based on relative distance comparison between sets of features. An iterative algorithm is used to train the model so that the distance between interesting pairs is smaller than the distance between irrelevant pairs. They scale their approach to larger problems using ensemble learning: a set of weak models is trained first; then an ensemble model is learned based on the weak models.

An interesting variation on the multicamera tracking problem is tracking from several unmanned aerial vehicles (UAVs). Behtke et al. [Bet07] developed a tracking algorithm for a fleet of UAVs. Each UAV is assumed to be able to identify a candidate for the target. Each UAV's navigation system provides an estimate of its position; the camera can estimate a direction vector to the target. The target's position is estimated from the set of measurements.

5.7 Use Cases and Workflows

Real-time video analysis opens up a wide range of new use cases for single cameras:

- Monitoring of scenes to detect activity or certain types of activity.
- Identification of people and vehicles, both by direct feature classification and by techniques such as license plate readers based on hyperresolution.
- Automated vehicles which provide features such as lane departure warnings, pedestrian excursion detection, and collision avoidance.

Networked smart cameras are the result of several developments: the creation of the Internet Protocol and wireless networking devices combined with low-cost cameras and video encoders allowed video to be sent over much longer distances than was possible with analog video. The application of several networked cameras extends the usefulness of many of these use cases and adds more:

- Tracking of the movement of large numbers of people and vehicles over large areas
- Monitoring and mapping using autonomous vehicles

The basic principles of exposure, focus, and tonal mapping are perhaps even more important to computer vision and automated analysis than is the case for photographs intended for viewing. The imaging parameters for the object of interest must be compatible with the parameters of the analysis algorithm; those imaging parameters must also be maintained as the subject moves through different lighting environments.

Most analytics require some type of extrinsic camera calibration to determine the camera's relationship to its environment. Applications with tightly controlled environments may not need background elimination, but cameras placed in more complex environments do need to separate objects of interest from other objects. Cameras on moving platforms—cars, unmanned aerial and water vehicles, etc.—require egomotion analysis. An increasing number of applications combine the results from multiple cameras; depending on the environment and the type of subject, camera results may be combined at different levels of abstraction ranging from low-level representations through initial classifications.

Further Reading

Books edited by Bobda and Velipasalar [Bob14] and Bhanu et al. [Bha11] discuss distributed smart camera networks.

Chapter 6
Photography and Cinematography

6.1 Introduction

Art and technology form a symbiotic relationship in many media. The history of drawing and painting, for example, is closely tied to the development of new materials. But the relationship between the technological means and artistic ends is perhaps no closer than in photography, an inherently technical medium. Photography is also unique in that it much more directly captures the natural world than do painting or sculpture. Although we have seen that photographs demand some amount of manipulation and interpretation, their quasi-realistic nature took years for the viewing public to digest and accept. Moving pictures raised this ambiguity to new heights by introducing time as a variable in our understanding of the presentation.

This short history outlines a few points in the development of still photography and cinematography. The succession of technological developments serves as a loose framework; aesthetic evolution is a critical aspect of our understanding of development of the form.

6.2 Photography

Early photography's development was, like many inventions, a story of parallel and competing inventors who had only partial knowledge of each other. The camera obscura had been known for centuries. Three key developments made photography possible. First, light-sensitive chemicals were identified. Second, chemicals were used to *develop* a *latent image* (the earliest experiments used extremely long exposures that caused the light-sensitive materials to change color so as to be directly visible). Third, chemical means were found to *fix* the image so it would not fade. In France, Joseph Niépce and Louis Daguerre worked first separately, with

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5_6

Fig. 6.1 A daguerreotype (Library of Congress [Unk63])

Niépce starting his work in the 1810s, then together starting in the 1820s. In England, William Fox Talbot developed a process using light-sensitive materials on paper. The photographic fixing problem was solved by Sir John Herschel.

Figure 6.1 shows a *daguerreotype*, the result of Daguerre's work. This process coated a piece of metal with light-sensitive materials. The image was viewed directly and could form a positive, but only when the photograph was held at the proper angle to reflect light off the metal and to the viewer. Daguerreotypes became extremely popular, thanks to their durability and low cost. Portraits, which once were the playthings of the rich, were now available to the broader public. However, the daguerreotype eventually fell out of favor because it could not be copied—each was unique. Fox Talbot's *calotype* process, in contrast, used the original exposure as a negative. A print was made by exposing a new piece of photographic material through the negative. However, because the negative was made of paper, the images were diffuse and indistinct.

Early photographic materials were not very sensitive to light. Exposures could require minutes. These long exposures made still life arrangements, buildings, and quiet natural scenes popular subjects. Photographs of people were also made, but they required the subject to stay still for the entire exposure. Early photographic materials were also not sensitive to all wavelengths of light; *panchromatic* film was introduced only at the turn of the twentieth century. The *collidon* process made use of an emulsion on a glass plate, which allowed much more detailed photographs. Early collidon processes required working with the plate while wet. The combination of glass and wet plates turned each photographer into a practicing chemist.

Nonetheless, photography was quickly applied to many arenas. Photographs of battle were taken of the Crimean War in 1855. The American Civil War was extensively photographed. Matthew Brady is the best known Civil War photographer his famous photograph of General Sherman is shown in Fig. 6.2 but other photographers also covered the war. Figure 6.3 shows President Abraham Lincoln and US Army officers at Antietam, Maryland; this photograph was taken by Alexander Gardner. Notice the relatively stiff poses to accommodate the required exposure times. This photograph was staged, but many other photographs of battle scenes were also taken.

Although early photographs appear stagey and artificial to us, early viewers considered photographs as literal representations. Newhall, for example, recounts how nineteenth-century viewers reacted strongly against a photograph of the death of a young girl and her attending family [New64]; they reacted to the staged photograph much more strongly than they would have to an equivalent painting.

Nineteenth-century photographers brought photography to remote locations, often preparing wet plates in difficult conditions. The US Civil War brought out photographers who made extensive and historically important photographs, sometimes immediately after a battle. T. Sherman; Fig. 6.3 shows President Lincoln and his generals at the site of a major. Later in the century, photographers documented the American West. Edward Curtis created an influential series of photographs of Native Americans and other scenes of the West. Panoramic photographers roamed the country in the late nineteenth and early twentieth centuries. They often created their portraits of towns and industrial sites on commission.

As photographic materials and equipment improved, photographs were increasingly useful in the capture of motion. Figure 6.4 shows *first flight*—the first powered, controlled flight. The Wright brothers hired a photographer to capture their flight to create a photograph that would accompany their patent application. After the flight, they asked "Did you get it?" The photographer was so stunned he wasn't sure if he took this photo. He had to develop the plate to know that he had remembered to fire the shutter.

George Eastman developed a form of roll film; early versions used a paper backing which was later improved to a film backing. He made use of this film in the Kodak, a handheld camera designed for amateur use. The aperture and shutter speed were fixed, requiring the photographer only to point and shoot. After shooting a roll of photos, the consumer mailed the camera back to Eastman's company; they developed and printed the film, loaded the camera with a new roll, and returned both the camera and photos. The Kodak was marketed under the slogan "You press the button, we do the rest." The Eastman system provided both ease of use and low cost and did much to popularize amateur photography.

Improved photographic processes also allowed photography to develop as an art. A key theme in art was the interplay between realism and abstraction. Alfred Stieglitz was influential in the early twentieth century as both a photographer and a curator. Stieglitz created the periodical *Camera Work* which became a very influential record of the artistic development of photography. Edward Steichen was an important photographer in the first half of the twentieth century. His work

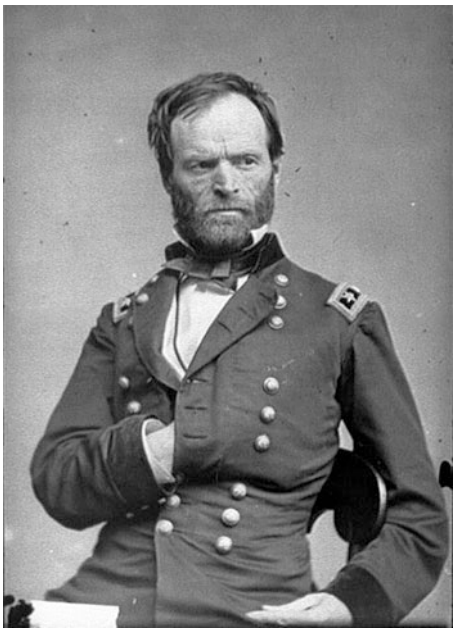


Fig. 6.2 Portrait of Maj. Gen. William T. Sherman, officer of the Federal Army (Library of Congress [Bra60])

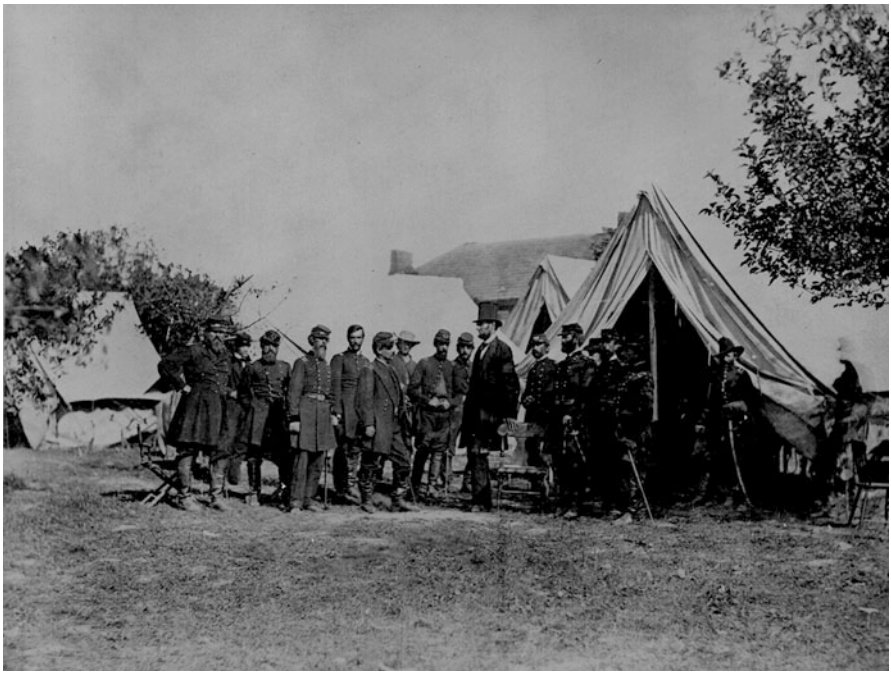


Fig. 6.3 Abraham Lincoln at Antietam (National Archives [Gar62])

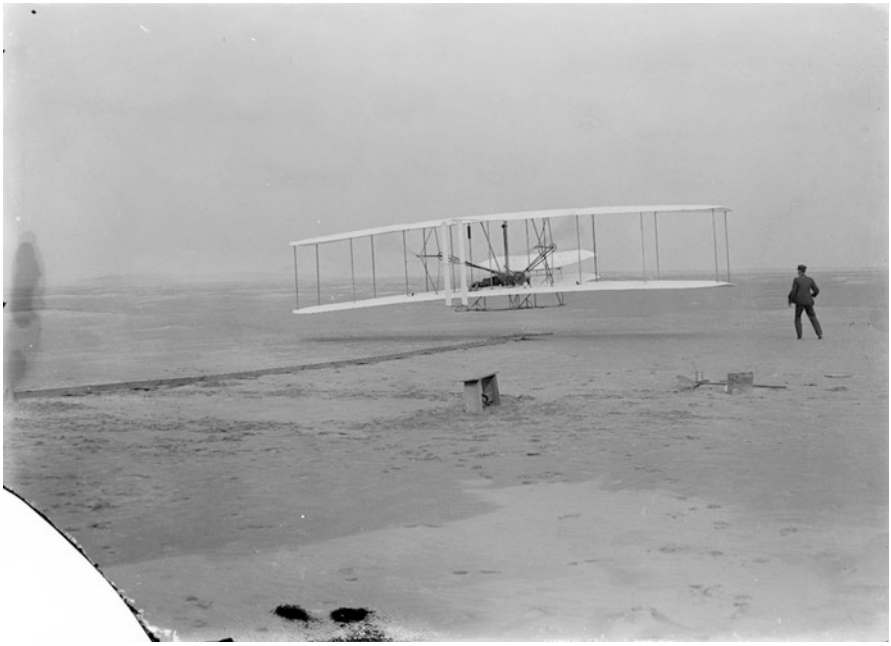


Fig. 6.4 Original Wright brothers' 1903 airplane ("Kitty Hawk") in first flight, December 17, 1903 at Kitty Hawk, N.C. Orville Wright at controls. Wilbur observing. (Library of Congress [Unk03])

included still lifes and urban compositions, but he was also an extremely successful commercial photographer. He also served as Director of Photography for the Museum of Modern Art, where he created the exhibit *The Family of Man*. Edward Weston made stunning photographs of both people and natural objects as abstract forms. Ansel Adams, a friend of Weston, concentrated on natural scenes with emphasis on the abstract use of their compositional forms and textures. Figure 6.5 shows one of Adams' photographs of the Tetons in Wyoming. Laszlo Moholy-Nagy was both a painter and photographer who made highly abstract photographs.

New processes allowed newspapers to print photographs. News photography added immediacy and often luridness to stories. Arthur Fellig worked under the name Wee Gee during the 1930s. His photographs of crime scenes, mostly on New York's Lower East Side, captured crime scenes in much the same way as they were seen by the police, thanks to his prompt arrival on crime scenes. His motto was "f/8 and be there." He published his works in a book titled *Naked City*.

The development of the 35 mm camera encouraged the development of new forms of photography of everyday life; the cameras were small enough to be used without attracting attention. Henri Cartier-Bresson became a master of the 35 mm form. He introduced the notion of the *decisive moment* at which the people and objects of a scene formed the best possible composition.

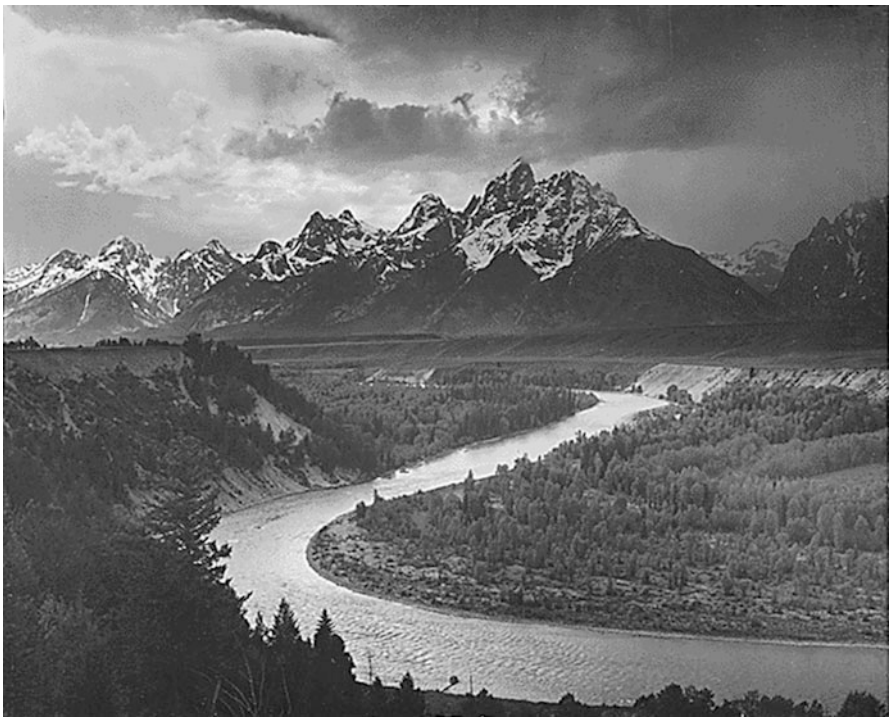


Fig. 6.5 "The Tetons—Snake River," Grand Teton National Park, Wyoming (National Archives [Ada41])

Photography became a tool for social commentary. Jacob Riis photographed the slums of New York in the early late 1880s and 1890s; his work was aided by the development of flash photography based on flash powder. He used these photographs as part of a campaign of newspaper stories and public speeches to campaign for reforms to improve slum conditions. During the Great Depression, the US Works Progress Administration employed many artists, including photographers. The photographers helped to create a *documentary* movement. Dorothea Lange and Walker Evans were key members of the WPA documentary team. During World War II, both Lange and Ansel Adams took photographs of the internment of Japanese Americans such as the photo of Fig. 6.6; Lange's photographs were shelved for many years by government officials who considered them to be too politically charged. The civil rights movement was a key subject for documentary photography as well as more traditional forms. Gordon Parks created a photo series for *Life* magazine which depicted scenes from the life of an African-American family in the segregated South. Parks went onto become a noted film director, most notably of the iconic film *Shaft*.

Space flight produced iconic and historic images. Astronaut Wally Schirra took a used Hasselblad camera on his Mercury flight; the photographs he took in orbit

Fig. 6.6 Manzanar from Guard Tower, view west (Sierra Nevada in background), Manzanar Relocation Center, California (Library of Congress [Ada43])

were so well received that photography became a central component of future missions. Figure 6.7 shows a photograph of Neil Armstrong on the moon; the footprints of Armstrong and Buzz Aldrin dot the landscape. We will discuss television coverage of space flight in the next section.

In the twenty-first century, the smartphone married photography and communication to create a range of new uses for photography. Participants broadcast photographs and videos of news events and social movements, sometimes live. People also photographed and broadcast their own illegal actions on social media. Early smartphone cameras were simple and provided only low-quality images. Modern smartphones use high-resolution sensors enhanced with advanced computation.

6.3 Cinematography

Cinematic artists have displayed varying interpretations of the roles of director and cinematographer. Some directors concentrate on the actors and leave photographic decisions to the cinematographer. On the other hand, John Frankenheimer declared that "the director decides what is in the frame," and Stanley Kubrick effectively acted as his own cinematographer. Decisions about lighting are one key area of

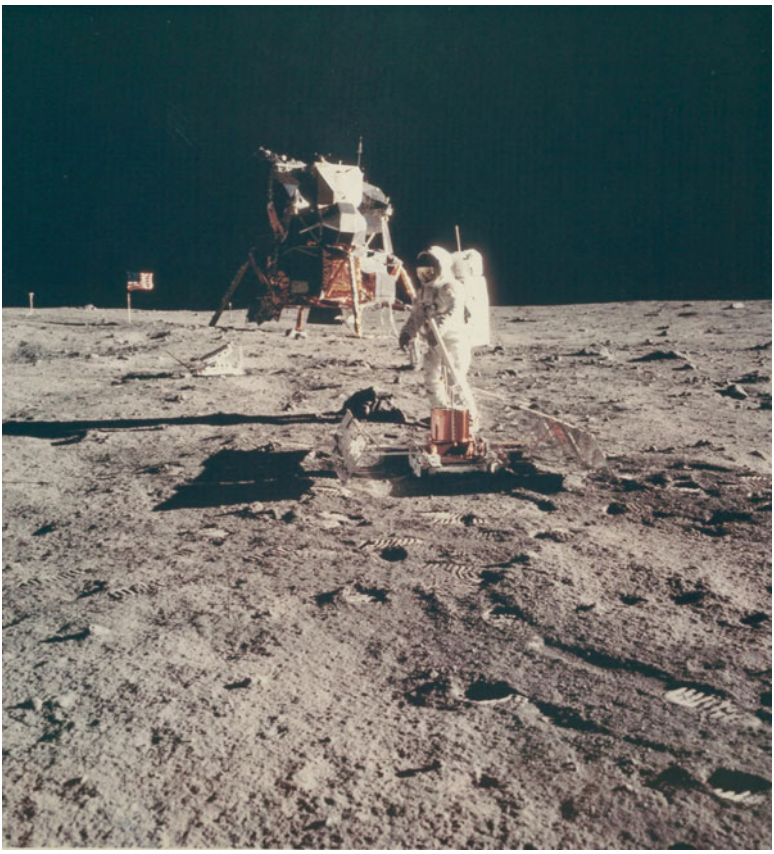


Fig. 6.7 Neil Armstrong on the moon (NASA [Ald69])

concern for the cinematographer. Perhaps no better example exists of the influence of lighting on the look of a film than *The Godfather*. Cinematographer Gordon Willis, ASC, chose to light Marlon Brando's face from above. The shadows cast upon Brando's eyes conveyed the lurking menace presented by Brando's character.

Eadweard Muybridge made pioneering photographic records of a moving horse in 1878. The image sequence was captured using a series of cameras placed along a track; the horse broke a wire at each camera location to fire its shutter. The image sequence was taken to settle a bet made by Leland Stanford as to whether all four of a horse's hooves were off the ground simultaneously. The experiments were conducted at what is now Stanford University.

The invention of the motion picture is generally credited to Thomas Edison. He worked on the problem of capturing and showing movement starting in 1889 and demonstrated it in 1893. The camera captured image sequences on flexible roll film, which was relatively novel at the time. A mechanism moved a portion of the film in front of the image frame, then paused, while a rotating shutter opened to expose the

image. Edison built a stage known as the Black Maria at his West Orange, New Jersey, laboratory; the stage rotated to allow it to follow the sun. Early movies were very short and almost anything captured in motion was attractive to audiences. The first motion picture copyrighted in the United States was *Fred Ott's Sneeze*, a film of a few seconds long whose content is perfectly captured by its title.

Edison invented the motion picture but did not invent the projector. His kinetoscope consisted of an eyepiece and lenses through which a rotating set of cards was viewed. The motion picture projector was invented by the brothers Auguste and Louis Lumi'ere. The French word *cinema* pays tribute to their contribution. They showed their first motion picture in 1895. They also invented Autochrome, the first useful color photographic process.

The form of the motion picture developed gradually over the first 20 years. Films became longer and told increasingly more complex and sophisticated stories. Genres also emerged. *The Great Train Robbery* was made in 1903. The film tells the story of a pair of bandits who stop a train, board it, terrorize the train employees and passengers, and steal the contents of its safe. The thieves are then chased down by a posse. The closing shot shows one of the bandits shooting directly at the camera. This film is regarded as the first Western.

By the early 1910s, the feature film had emerged as the long form of cinema. A milestone feature film was *The Birth of a Nation* directed by D. W. Griffith and released in 1915. This film was a morally reprehensible work depicting the formation of the white supremacist Ku Klux Klan, but it also codified a number of cinematic techniques.

During the 1910s and 1920s, a number of cinematic techniques were developed, including the closeup and the dialog scene. As these techniques were introduced, audiences learned how to interpret them, allowing subsequent directors to use those techniques and build on them to create even more sophisticated sequences. The result was the development of the visual language of cinema. Because films were silent, this visual language was shared across the world. Films did rely on title cards interspersed to provide description. Ultimately, directors raced to make the first feature with no title cards,

The 1920s were perhaps the high point of cinematic comedy. The reigning comedic geniuses of the era included Charlie Chaplin (arguably the first global star), Buster Keaton, and Harold Lloyd. All three combined physical comedy with strong characterization. Chaplin's *The Gold Rush*, released in 1925, is a masterpiece. Among its many achievements, it introduced what would become a trope in cartoons—a hungry person imagining another person as a juicy, steaming roast chicken. Buster Keaton, The Great Stoneface, mixed impassive observation of chaos around him with impressive physical reactions of his own to create works that still seem modern and fresh. Harold Lloyd's glasses character used a gee-whiz persona as a front for daredevil feats. His *Safety Last* is a lasting commentary on modern life. Hal Roach was the most important producer of comedy shorts. Among other achievements, he paired Stan Laurel and Oliver Hardy. He lived long enough to be a guest on *Late Night with David Letterman*.

Edison tried to synchronize phonographs to motion pictures but eventually gave up. Lee De Forest, a radio pioneer, developed a synchronized sound system. But the film which changed the motion picture industry relied on a relatively primitive technique. *The Jazz Singer*, released in 1927, relied on records played at key points in the film; most of the film was silent. The film starred Al Jolson, a popular entertainer. It became a sensation and audiences demanded more sound films. Within 2 years, the entire industry had converted to sound production. The complexities of capturing synchronized sound kept film productions, which in the silent history had made extensive use of locations, into the studio for years to come.

The silent era film industry was very international. European studios introduced many innovations and made many popular films. The industry consolidated under the sound era. Although the introduction of the spoken word made films in some ways less portable, the much higher cost of sound production outweighed that problem. Hollywood became a global source for film; its rise was aided by the troubles in Europe. The Hollywood studios did make efforts to internationalize their products; Laurel and Hardy, for example, learned their lines phonetically in Spanish and produced dual language versions of every shot with the spoken word.

Nanook of the North (1922) was an early documentary feature. It told the story of an Inuit family living in northern Quebec. A great deal of the footage was staged and today would be considered something closer to a docudrama. However, it is popular in its initial release and today is considered a pioneering film.

The movie musical developed quickly in the early 1930s. Busby Berkeley became known for his complex musical numbers with large choruses of dancers. The dancers were often shot from above, forming geometric shapes that shifted as they moved. Berkeley also directed a series of films costarring Judy Garland and Mickey Rooney and went on to create the Esther Williams water spectaculars. His gift can best be described as the ability to visually portray music.

Fred Astaire, a successful Broadway dancer, moved to Hollywood in the 1930s. His films of that decade with Ginger Rogers as his partner are regarded as classics of the form. Astaire was a meticulous craftsman who insisted that the dances be shot at full body length without close-ups.

The Western was a fixture of both the silent and sound eras and evolved into a genre that was uniquely suited to film. William S. Hart was the foremost movie cowboy of the silent era; his characters were portrayed as noble and honest. *The Big Trail* (1930), directed by Raoul Walsh, was an early talkie Western that was also shot in an early widescreen processes using 70 *mm* film; it featured John Wayne in his first starring role. The director John Ford created the visual template for the Western as he settled on Monument Valley, located in the Navajo Nation Reservation on the Arizona-Utah border, as the location for many of his films. Ford's first film in Monument Valley was *Stagecoach* (1939), which made a star of John Wayne. *The Searchers* (1956) also starred Wayne but in a much darker and more melancholy story that is considered one of the greatest examples of the form.

Color photography, unlike sound, took years to develop and to effectively introduce into motion picture production. Technicolor was the most successful color motion picture process, but the process evolved over several major steps.

An early form was used in several pictures in the early 1920s, but this form could not capture or display a full-color gamut. Full-color Technicolor was introduced in the late 1920s, but the Great Depression slowed its introduction. The Disney cartoon *Flowers and Trees* from 1932 was a relatively early use of the three-strip Technicolor process.

Electronic television was invented by Philo Farnsworth in 1927. Television also took years to develop technically and even longer to come into common use. By the late 1930s, television technology had advanced to the point that it was demonstrated at the New York World's Fair. However, World War II delayed the introduction of regularly scheduled television programming until the late 1940s.

Early television production relied on live editing. The first videotape recorder was not introduced until 1956 [Amp17] and was used sparingly for years. Feeds from each camera were brought to a console operated by a director and technicians. The director switched the output signal between cameras in order to create the shot sequence for the show.

The 1950s are known as the Golden Age of Television due to the large number of high-quality programs and the large numbers of talented performers and directors who emerged during those years. *Playhouse 90* was one of several highly acclaimed dramatic series which aired full-length dramas performed live, often with sophisticated sets and complex camerawork. John Frankenheimer, Arthur Penn, and Franklin J. Schaffner were acclaimed television directors who went onto successful careers in film. Paddy Chayefsky wrote the acclaimed television drama *Marty* and went onto write *Network* and many other films. Rod Serling wrote television dramas, including an early dramatization of an airplane hijacking and then went on to create the classic *The Twilight Zone* as well as the screenplay for *Planet of the Apes*. *The Twilight Zone* used fantasy as a framing device for the social commentary themes which Serling had developed more overly in his earlier work.

I Love Lucy, which aired from 1951 to 1957, was a seminal program in the history of television. Lucille Ball was a gifted comedic actress who possessed both perfect timing and physical comedy skills that embodied ridiculousness. Her appearance at the dawn of television helped define comedy and the role of women in television. Her husband Desi Arnaz played a supporting role on the show that required him largely to react to his wife's ridiculousness. Behind the camera, Arnaz was a brilliant producer who created key forms in the emerging medium. He insisted that the show be shot in front of a live audience like a play. This was a bold decision given the technical challenges of television production at the time. He hired cinematographer Karl Freund ASC to shoot the show. Together they developed the three-camera setup—master and two close-ups—that still form the backbone of television. Freund also developed a flat lighting style that allowed them to shoot from all angles simultaneously; in contrast, cinematic productions adjusted the lighting for each angle. Arnaz also decided to spend the extra money to shoot the show on film. This gave him a much higher-quality record than was possible with the kinescopes of the time. The result was the enablement of the rerun, which both popularized key shows through repetition and magnified the earnings of popular shows. *I Love Lucy* is said to have run continuously around the world for

decades; many remarked that the first messages from Earth received by aliens on other worlds would be pictures of Lucille Ball.

Television news also emerged as a distinct medium with Edward R. Murrow, already famous for his work in radio, creating programs such as *Person to Person*. A critical event in the development of television news was the coverage of the Army-McCarthy hearings of the US Senate, which were prompted by the allegations made by Senator Joseph McCarthy of Communist infiltration of the US government. The hearings were covered live for several weeks by both the ABC and DuMont networks. Presentation of the behavior of McCarthy, along with newspaper coverage, resulted in a shift in public opinion against McCarthy. Another key event in the development of television news was the coverage of the Kennedy-Nixon debates in 1960. Nixon refused to wear makeup for the first debate and presented a poor visual appearance; Kennedy, in contrast, was much more telegenic.

Spaceflight provided major television events throughout the 1960s. Live coverages of key events such as launch and reentry were standard procedure. Apollo 7, which remained in Earth orbit, provided the first live television broadcast from space. Two months later, Apollo 8 broadcast a program from lunar orbit on Christmas Eve, 1968, which received the largest audience of any television broadcast up to that time. Apollo 11 broadcast live Neil Armstrong's first step onto the moon.

In cinema, the French *Nouvelle Vague* (*New Wave*) was created by a generation of young directors starting in the late 1950s. Some of them first made their mark as film critics before moving onto making their own films. Jean-Luc Godard and Francois Truffaut are two members of the New Wave who created major bodies of work. Italian cinema also blossomed after World War II. Italian neorealism, embodied in works such as Roberto Rossellini's *Open City* (1945) and Vittorio De Sica's *Bicycle Thieves* (1948), was a response to difficult conditions after the war. Federico Fellini started in a neorealist style and then moved onto a more fantastical style.

In the late 1960s and 1970s, Hollywood studios hired a generation of young directors in an attempt to fight increasing competition from television. These directors made use of a range of innovative techniques, some of them borrowed from influential European directors. Key examples of the period were Francis Ford Coppola's *The Godfather* (1972), which became a huge hit, and its sequel *The Godfather: Part II*, (1974), also a huge hit. The downbeat themes of the latter film would have been unthinkable in a studio film a decade before. Their experiments in technique were also aided by technical advances, such as zoom lenses and faster film stocks.

Documentary film evolved in the postwar period to embrace a style known as *cinéma vérité*. Filmmakers made use of smaller, more portable equipment to attempt a less intrusive style of filming. They also edited their footage in a style that less overtly imposed a narrative. Major American practitioners of the form included Albert and David Maysles and D. A. Pennebaker. The television documentary *An American Family* (1976) portrayed 7 months in the life of a California

family. The filmmakers expected to capture a slice of life; instead, the film ended with the husband and wife deciding to divorce and the son coming out as gay. That series is now widely regarded as a progenitor of reality TV. Albert Brooks' *Real Life* (1979) parodies *An American Family* and itself widely regarded as presaging some of the seedier aspects of modern reality programming. It contains a montage that is both an excellent example and a brilliant parody of the form.

The first film created entirely by computer graphics was *The Last Starfighter* (1984). However, the film that made CGI a key force in major motion pictures was *Terminator II: Judgment Day* (1991). That film's liquid terminator character demonstrated both how convincing computer graphics could be and how CGI could be used in service of the story.

The Steadicam system was introduced in the 1970s. It allowed a camera operator to carry the camera on a mount attached to the operator's body and to make very smooth camera moves. Sequences in two films, both from 1976, demonstrated Steadicam's capabilities: Rocky Balboa running up the steps in Philadelphia in *Rocky* and Stanley Kubrick's shot behind a boy's tricycle in *The Shining*.

Two films reinvented the chase scene. Earlier films had shot chases either on stages using special effects or with relatively simple camera setups. Hitchcock's *North by Northwest* (1959) presented Cary Grant being chased across an Illinois cornfield by a PT-17 crop duster. Hitchcock explained to Francois Truffaut [Tru85] that he wanted to subvert the traditional suspense dynamic of the character underneath a streetlamp on a darkened street waiting for an event to occur. Grant's character, in contrast, is told to travel to an isolated rural area to wait for an unspecified event. Peter Yates' *Bullitt* (1969) features what is still one of the greatest car chases in cinema. It was filmed on the streets of San Francisco using high-performance cars. Much of the driving was performed by stunt drivers, but Steve McQueen drove the Mustang for close-ups; studio-bound films had been unable to present the star as the subject of such clear and intense jeopardy. *Bullitt* won the Academy Award for best editing, thanks to Frank Keller's ability to clearly convey the geography and plot of the chase while maintaining its kinetic energy.

Hong Kong filmmakers redefined the action film. *Enter the Dragon* (1973) made an international star of Bruce Lee and depicted fight sequences using shots that clearly showed off the moves of the fighters. Jackie Chan's films were famous for his incredible stunt work; his films ended with outtakes of the stunts, often including shots of him being taken to the hospital. John Woo created a series of films that depicted gunfights almost as ballets; *The Killer* (1989) is a prime example of his work.

Akira Kurosawa was the most famous Japanese director of the postwar film. His work covered a range of themes and periods. But the samurai ethic was an important influence on his thought and the American Western a significant influence on his work. *The Seven Samurai* (1954) tells the story of a group of unemployed fighters hired by a village to protect them from bandits. The samurai are loners, much like the traditional cowboy. That film went onto influence the American Western when it was remade as *The Magnificent Seven* (1960). His film *Yojimbo* (1961) became the basis for *A Fistful of Dollars* (1964).

The combination of broadcast, cable, and streaming services in the early twenty-first century created huge demand for content. In 2015, FX network CEO John Landgraf predicted that "peak TV"—meaning the largest number of scripted programs in production—would occur in 2015 or 2016 [Ada15]. Landgraf later updated his prediction; his network released a study in 2016 that counted 455 scripted series in production in the United States in 2016, up from 421 in 2015 and 266 in 2011 [Zuc16].

Further Reading

Beaumont Newhall's *The History of Photography* [New64] is a standard reference which describes the interplay between technical and artistic developments.

Mast [Mas92] gives a comprehensive survey of the history of motion pictures. Bordwell [Bor85] analyzes the cinematic forms used to tell stories. *The Parade's Gone By* [Bro68] is an important work on silent-era Hollywood. Louise Brooks was a silent film star; her autobiography *Lulu in Hollywood* [Bro82] provides a glimpse into work and life in the silent movie period and is also very well written. *American Cinematographer* provides a wealth of interviews and articles. Francois Truffaut's *Hitchcock* [Tru85] is not only a discussion between two master filmmakers but benefits from Truffaut's journalistic training. Avclub.com's series *A History of Violence* explores the history of the action film in detail.

References

[Ada41] Ansel Adams, "The Tetons---Snake River, Grand Teton National Park, Wyoming," 1941-42, Series Ansel Adams Photographs of National Parks and Monuments, 1941---1942, ARC Identifier 519904, NAIL Control Number NWDNS-79-AA-G01.

[Ada43] Ansel Adams, "Manzanar from Guard Tower, view west (Sierra Nevada in background), Manzanar Relocation Center, California," 1943, Reproduction Number: LC-DIG-ppprs-00200. Call Number: LC-A351-3-M-4-Bx [P&P]. Library of Congress Prints and Photographs Division Washington, D.C. 20540 USA

[Ada02A] Ansel Adams, *The Camera, The Ansel Adams Photography Series 1*, New York: Little, Brown and Company, 2002.

[Ada02B] Ansel Adams, *The Negative, The Ansel Adams Photography Series 2*, New York: Little, Brown and Company, 2002.

[Ada02C] Ansel Adams, *The Print, The Ansel Adams Photography Series 3*, New York: Little, Brown and Company, 2002.

[Ada15] Erik Adams, "TCA roundup: '2015 or 2016 will represent peak TV in America'", avclub.com, Aug 7, 2015, http://www.avclub.com/article/tca-roundup-2015-or-2016-will-represent-peak-tv-am-223558

[Aga04] Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, and Michael Cohen. 2004. Interactive digital photomontage. In *ACM SIGGRAPH 2004 Papers* (SIGGRAPH '04), Joe Marks (Ed.). ACM, New York, NY, USA, pp. 294–302. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/1186562.1015718

[Ahm74] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," in *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, Jan 1974. doi: https://doi.org/10.1109/T-C.1974.223784

[Ald69] Buzz Aldrin, "Photograph of Neil Armstrong on the Moon,"*From RG: 255, Project Files on the Early Apollo Surface Experiments Package (EASEP)*. This item is a photograph of Neil Armstrong on the moon. National Archives Identifier: 4957965. Creator: National Aeronautics and Space Administration. Manned Spacecraft Center. Science and Applications Directorate. 1/1967-2/17/1973

[Ald88] Aldus and Microsoft, *TIFF Revision 5.0*, Aug. 8, 1988.

[All85] Ross R. Allen, John D. Meyer, and William R. Knight, "Thermodynamics and hydrodynamics of thermal ink jet printers," in *Hewlett-Packard Journal*, vol. 36, no. 5, May 1985, pp. 21–27.

[Amp17] Ampex, "Ampex History," http://www.ampex.com/ampex-history/, accessed Jan. 26, 2017.

M. Wolf, *Smart Camera Design*, https://doi.org/10.1007/978-3-319-69523-5

[ARM09] ARM, *Cortex-A9 MPCore Technical Reference Manual*, Revision r2p0, 2009.
[Arn74] Rudolph Arnheim, *Art and Visual Perception: A Psychology of the Creative Eye, The New Version*, Berkeley and Los Angeles CA: University of California Press, 1974.
[B16] Benjamin B, "Cutting-edge clarity," in *American Cinematographer*, vol. 97, no. 12, Dec 2016, pp. 34–49.
[Bay75] Bryce E. Bayer, "Color imaging array," U. S. Patent 3,971,065, Mar 5, 1975.
[Bay06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features," in *ECCV 2006: European Conference on Computer Vision, LCNS*, vol. 3951, Springer, 2006, pp. 404–417.
[Bel92] Cynthia S. Bell, "Contrast-based autofocus mechanism," U. S. Patent 5,170,202, Dec 8, 1992.
[Bel02] Cynthia S. Bell, Edward P. Tomaszewski, Amy E. Hansen, and Kannan Raj, "Determining a final exposure setting automatically for a solid state camera without a separate light metering circuit," U. S. Patent 6,486, 915, Nov 26, 2002.
[Bet07] Bethke B., Valenti M., How J. (2007) Cooperative Vision Based Estimation and Tracking Using Multiple UAVs. In: Pardalos P.M., Murphey R., Grundel D., Hirsch M.J. (eds) Advances in Cooperative Control and Optimization. Lecture Notes in Control and Information Sciences, vol. 369. Springer, Berlin, Heidelberg
[Bha85] Eldurkar V. Bhaskar and J. Stephen Alden, "Development of the thin-film structure for the ThinkJet printhead," in *Hewlett-Packard Journal*, vol. 36, no. 5, May 1985, pp. 27–33.
[Bha11] Bir Bhanu, Chinya V. Ravishankar, Amit K. Roy-Chowdhury, Hamid Aghajan, and Demetri Terzopoulos, eds., *Distributed Video Sensor Networks*, Springer, 2011.
[Bob14] Christophe Bobda and Senem Velipasalar, eds., *Distributed Embedded Smart Cameras: Architectures, Design, and Applications*, Springer, 2014.
[Bor85] David Bordwell, *Narration in the Fiction Film*, Madison WI: University of Wisconsin Press, 1985.
[Bov13] A. C. Bovik, "Automatic Prediction of Perceptual Image and Video Quality," in *Proceedings of the IEEE*, vol. 101, no. 9, Sept 2013, pp. 2008–2024.
[Boy70] W. S. Boyle and G. E. Smith, "Charge Coupled Semiconductor Devices," in *Bell System Technical Journal*, vol. 49, no. 4, Apr 1970, pp. 587–593.
[Bra65] Matthew Brady, "Portrait of Maj. Gen. William T. Sherman, officer of the Federal Army," between 1860 and 1865. Brady National Photographic Art Gallery, Reproduction Number LC-DIG-cwpb-07136, Call Number LC-B813- 6454 A, Library of Congress Prints and Photographs Division Washington, D.C. 20540 USA http://hdl.loc.gov/loc.pnp/pp.print
[Bra00] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844–851, Aug 2000. doi: https://doi.org/10.1109/34.868685
[Bra60] Brady National Photographic Art Gallery (Washington DC), Maj. Gen. William Tecumseh Sherman; half-length, seated, ca. 1860- ca. 1865, 111-B-1769, National Archives Identifier 526708.
[Bro68] Kevin Brownlow, *The Parade's Gone By*, Berkeley CA: University of California Press, 1968.
[Bro82] Louise Brooks, Lulu in Hollywood, New York: Alfred A. Knopf, 1982.
[Bro03] M. Brown and D. G. Lowe, "Recognising panoramas," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1218–1225 vol. 2. doi: https://doi.org/10.1109/ICCV.2003.1238630
[Bur83] P. Burt and E. Adelson, "The Laplacian Pyramid as a Compact Image Code," in *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, Apr 1983. doi: https://doi.org/10.1109/TCOM.1983.1095851
[But10] Richard Butler, "Exclusive: Fujifilm's phase detection system explained," dpreview.com, Aug 5, 2010.

[Car78] Ingrid Carlbolm and Joseph Paciorek, "Planar geometric projections and viewing transformations," in *Computing Surveys*, vol. 10, no. 4, Dec 1978, pp. 465–504.

[Cha01] T. H. Chang and S. Gong, "Tracking multiple people with a multi-camera system," in *Proceedings 2001 I.E. Workshop on Multi-Object Tracking*, Vancouver, BC, 2001, pp. 19–26. doi: https://doi.org/10.1109/MOT.2001.937977

[Che01] N. Ng Kuang Chern, Poo Aun Neow and M. H. Ang, "Practical issues in pixel-based autofocusing for machine vision," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001, pp. 2791–2796, vol. 3. doi: https://doi.org/10.1109/ROBOT.2001.933045

[Cho91] Nam Ik Cho and San Uk Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," in *IEEE Transactions on Circuits and Systems*, vol. 38, no. 3, pp. 297–305, Mar. 1991. doi: https://doi.org/10.1109/31.101322

[Cho09] Junguk Cho, Shahnam Mirzaei, Jason Oberg, and Ryan Kastner. 2009. Fpga-based face detection system using Haar classifiers. In *Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays* (FPGA '09). ACM, New York, NY, USA, 103–112. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/1508128.1508144

[Cip10] Camera and Imaging Products Association, *Design rule for Camera File system*" *DCF Version 2.0 (Edition 2010)*, Standardization Committee, Camera and Imaging Products Association, Apr 26, 2010.

[Com03] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003. doi: https://doi.org/10.1109/TPAMI.2003.1195991

[Cox66] Arthur Cox, *Photographic Optics: A Modern Approach to the Technique of Definition*, thirteenth edition, London and New York: Focal Press, 1966.

[Deb97] Paul E. Debevec and Jitendra Malik. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '97). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 369–378. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/258734.258884

[Dec98] S. Decker, D. McGrath, K. Brehmer and C. G. Sodini, "A 256×256 CMOS imaging array with wide dynamic range pixels and column-parallel digital output," in *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 2081–2091, Dec 1998. doi: https://doi.org/10.1109/4.735551

[Dev06] Dhanya Devarajan, Richard J. Radke, and Haeyong Chung. 2006. "Distributed metric calibration of ad hoc camera networks," in *ACM Transactions on Sensor Networks,* vol. 2, no. 3, Aug 2006, pp. 380–403. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/1167935.1167939

[Dic95] A. Dickinson, B. Ackland, E. S. Eid, D. Inglis and E. R. Fossum, "Standard CMOS active pixel image sensors for multimedia applications," in *Proceedings Sixteenth Conference on Advanced Research in VLSI*, Chapel Hill, NC, 1995, pp. 214–224. doi: https://doi.org/10.1109/ARVLSI.1995.515622

[Dou08] Arnaud Doucet and Adam M. Johansen, "A tutorial on particle filtering and smoothing: fifteen years later," version 1.1, Dec 2008.

[Dut96] S. Dutta and W. Wolf, "A flexible parallel architecture adapted to block-matching motion-estimation algorithms," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 1, pp. 74–86, Feb 1996. doi: https://doi.org/10.1109/76.486422

[Elk04] O. Elkhalili, O. M. Schrey, P. Mengel, M. Petermann, W. Brockherde and B. J. Hosticka, "A 4×64 pixel CMOS image sensor for 3-D measurement applications," in *IEEE Journal of Solid-State Circuits*, vol. 39, no. 7, pp. 1208–1212, July 2004. doi: https://doi.org/10.1109/JSSC.2004.829927

[Fed06] D. Fedorov, B. Sumengen and B. S. Manjunath, "Multi-Focus Imaging using Local Focus Estimation and Mosaicking," in *2006 International Conference on Image*

Processing, Atlanta, GA, 2006, pp. 2093–2096. doi: https://doi.org/10.1109/ICIP.2006.312820

[Fey10] Richard P. Feynman, Robert B. Leighton, and Matthew Sands, *The Feynman Lectures on Physics, Volume I: Mainly Mechanics, Radiation and Heat*, Millenium Edition, New York: Basic Books, 2010.

[Fis81] Martin A. Fischler and Robert C. Bolles. 1981. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6 (June 1981), pp. 381–395. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/358669.358692

[Fli95] M. Flickner *et al*., "Query by image and video content: the QBIC system," in *Computer*, vol. 28, no. 9, pp. 23–32, Sep 1995. doi: https://doi.org/10.1109/2.410146

[Fli12] Flir Systems, Inc., *The Ultimate Infrared Handbook for R&D Professionals*, Flir Systems, Inc., 2012.

[Fol96] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes, *Computer Graphics: Principles and Practice, second edition in C*, Menlo Park CA: Addison-Wesley, 1996.

[For83] T. Fortmann, Y. Bar-Shalom and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," in *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, Jul 1983. doi: https://doi.org/10.1109/JOE.1983.1145560

[Fow75] Grant R. Fowles, *Introduction to Modern Optics*, *second edition*, Mineola NY: Dover Publications, 1975.

[Gar62] Photographed by Alexander Gardner. President Lincoln visiting the battlefield at Antietam, Md., Oct 3, 1862. General McClellan and 15 members of his staff are in the group. 165-SB-23. National Archives Identifier: 533297

[Ghe06] N. Gheissari, T. B. Sebastian and R. Hartley, "Person Reidentification Using Spatio-temporal Appearance," in *2006 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, pp. 1528–1535. doi: https://doi.org/10.1109/CVPR.2006.223

[Gof12] S. Goferman, L. Zelnik-Manor and A. Tal, "Context-aware saliency detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 1915–1926, Oct 2012. doi: https://doi.org/10.1109/TPAMI.2011.272

[Gon17] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, fourth edition, Pearson, 2017.

[Gow07] R. D. Gow, David Renshaw, Keith Findlater, Stuart J. McLeod, John Hart, and Robert L. Hicol "A Comprehensive Tool for Modeling CMOS Image-Sensor-Noise Performance," in *IEEE Transactions on Electron Devices*, vol. 54, no. 6, pp. 1321–1329, June 2007. doi: https://doi.org/10.1109/TED.2007.896718

[Gra72] E. M. Granger and K. N. Cupery, "An optical merit function (SQF), which correlates with subjective image judgements," in *Photographic Science and Engineering*, vol. 16, no. 3, May–June 1972, pp. 221–238.

[Gra10] M. Granados, B. Ajdin, M. Wand, C. Theobalt, H. P. Seidel and H. P. A. Lensch, "Optimal HDR reconstruction with linear digital cameras," in *2010 I.E. Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010, pp. 215–222. doi: https://doi.org/10.1109/CVPR.2010.5540208

[Gre50] Allen R. Greenleaf, *Photographic Optics*, New York: The MacMillan Company, 1950.

[Gun05] Bahadir K. Gunturk, John Glotzbach, Yucel Altunbasak, Ronald W. Schafer, and Russel M. Mersereau, "Demosaicking: color filter array interpolation," in *IEEE Signal Processing Magazine,* Jan 2005, pp. 44–54.

[Guo92] J. I. Guo, C. M. Liu and C. W. Jen, "The efficient memory-based VLSI array designs for DFT and DCT," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 39, no. 10, pp. 723–733, Oct 1992. doi: https://doi.org/10.1109/82.199898

[Ham92] Eric Hamilton, *JPEG File Interchange Format, Version 1.02*, Sept 1, 1992.

[Har97] Richard I. Hartley, "In defense of the eight-point algorithm," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, June 1997, pp. 580–593.

[Har03] Richard Hartley and Andrew Zisserman, *Multiple View Geometry in Computer Vision*, second edition, Cambridge: Cambridge University Press, 2003.

[Har07] H. Hariharan, A. Koschan and M. Abidi, "Multifocus Image Fusion by Establishing Focal Connectivity," in *2007 I.E. International Conference on Image Processing*, San Antonio, TX, 2007, pp. III - 321–III - 324. doi: https://doi.org/10.1109/ICIP.2007.4379311

[Has10] S. W. Hasinoff, F. Durand and W. T. Freeman, "Noise-optimal capture for high dynamic range photography," in *2010 I.E. Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010, pp. 553–560. doi: https://doi.org/10.1109/CVPR.2010.5540167

[Has16] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. 2016. "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics,* vol. 35, no. 6, Article 192 (Nov 2016), pp. 12. doi: https://doi-org.prx.library.gatech.edu/10.1145/2980179.2980254

[Heb08] Rene Helbing, "Still image stabilization suitable for compact camera environments," U. S. Patent Application Publication US 2008/0030587 A1, Feb 7, 2008.

[Hig52] G. C. Higgins and L. A. Jones, "The Nature and Evaluation of the Sharpness of Photographic Images," in *Journal of the Society of Motion Picture and Television Engineers*, vol. 58, no. 4, pp. 277–290, Apr 1952. doi: https://doi.org/10.5594/J01196

[Hil01] Frederick S. Hillier and Gerald J. Lieberman, *Introduction to Operations Research*, seventh edition, New York: McGraw-Hill, 2001.

[Hir08] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb 2008. doi: https://doi.org/10.1109/TPAMI.2007.1166

[Hon17] Christiana Honsberg and Stuart Bowden, "Optical Properties of Silicon," pveducation.org, http://pveducation.org/pvcdrom/materials/optical-properties-of-silicon, accessed June 8, 2017

[Hua00] J. Huang, A. B. Lee and D. Mumford, "Statistics of range image," Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000, Hilton Head Island, SC, 2000, pp. 324–331 vol. 1.

[Hou78] Hsieh Hou and H. Andrews, "Cubic splines for image interpolation and digital filtering," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, pp. 508–517, Dec 1978. doi: https://doi.org/10.1109/TASSP.1978.1163154

[Hou07] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," in *2007 I.E. Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007, pp. 1–8. doi: https://doi.org/10.1109/CVPR.2007.383267

[Hug08] C. Hughes, M. Glavin, E. Jones and P. Denny, "Review of geometric distortion compensation in fish-eye cameras," in *IET Irish Signals and Systems Conference (ISSC 2008)*, Galway, 2008, pp. 162–167. doi: https://doi.org/10.1049/cp:20080656

[Ira94] M. Irani, B. Rousso and S. Peleg, "Recovery of ego-motion using image stabilization," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 454–460. doi: https://doi.org/10.1109/CVPR.1994.323866

[Ish03] C. Ishii, Y. Sudo and H. Hashimoto, "An image conversion algorithm from fish eye image to perspective image for human eyes," in *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, 2003, pp. 1009–1014 vol. 2. doi: https://doi.org/10.1109/AIM.2003.1225480

[Itt98] L. Itti, C. Koch and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, Nov 1998. doi: https://doi.org/10.1109/34.730558

[Jav05] O. Javed, K. Shafique and M. Shah, "Appearance modeling for tracking in multiple non-overlapping cameras," in *2005 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 26–33 vol. 2. doi: https://doi.org/10.1109/CVPR.2005.71

[JEI02] Technical Standardization Committee on AV & IT Storage Systems and Equipment, *Exchangeable image file format for digital still cameras: Exif Version 2.2*, Japan Electronics and Information Technology Industries Association, JEITA CP-3451, Apr, 2002.

[Jep03] A. D. Jepson, D. J. Fleet and T. F. El-Maraghi, "Robust online appearance models for visual tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, Oct 2003. doi: https://doi.org/10.1109/TPAMI.2003.1233903

[Kan05] Jinman Kang, I. Cohen, G. Medioni and Chang Yuan, "Detection and tracking of moving objects from a moving platform in presence of strong parallax," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, 2005, pp. 10–17, vol. 1. doi: https://doi.org/10.1109/ICCV.2005.72

[Kap85] S. Kappagantula and K.R. Rao, "Motion Compensated Predictive Interframe Coding," in *IEEE Transactions on Communications*, vol. 33, no. 9, pp. 1011–1015, Sept 1985.

[Kav00] S. Kavadias, B. Dierickx, D. Scheffer, A. Alaerts, D. Uwaerts and J. Bogaerts, "A logarithmic response CMOS image sensor with on-chip calibration," in *IEEE Journal of Solid-State Circuits*, vol. 35, no. 8, pp. 1146–1152, Aug 2000. doi: https://doi.org/10.1109/4.859503

[Ket99] V. Kettnaker and R. Zabih, "Bayesian multi-camera surveillance," in *Proceedings. 1999 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Fort Collins, CO, 1999, pp. 259, vol. 2. doi: https://doi.org/10.1109/CVPR.1999.784638

[Key81] R. Keys, "Cubic convolution interpolation for digital image processing," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 6, pp. 1153–1160, Dec 1981. doi: https://doi.org/10.1109/TASSP.1981.1163711

[Kha03] S. Khan and M. Shah, "Consistent labeling of tracked objects in multiple cameras with overlapping fields of view," in *IEEE Transactions on PAMI*, vol. 25, no. 10, pp. 1355–1360, Oct 2003.

[Kim08] Sujung Kim, Wook-joong Kim and Seong-Dae Kim, "Automatic white balance based on adaptive feature selection with standard illuminants," in *2008 15th IEEE International Conference on Image Processing*, San Diego, CA, 2008, pp. 485–488. doi: https://doi.org/10.1109/ICIP.2008.4711797

[Kim09] Honggab Kim, Romberg, J.; Wolf, W., "Multi-camera tracking on a graph using Markov chain Monte Carlo," in *Third ACM/IEEE International Conference on Distributed Smart Cameras, 2009. ICDSC 2009,* vol., no., pp. 1–8, Aug 30 2009–Sept 2 2009.

[Kim10] Honggab Kim and Marilyn Wolf, "Distributed tracking in a large-scale network of smart cameras," in Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras, ACM Press, 2010, pp. 8–16.

[Kle01] S. Kleinfelder, SukHwan Lim, Xinqiao Liu and A. El Gamal, "A 10000 frames/s CMOS digital pixel sensor," in *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, Dec 2001. doi: https://doi.org/10.1109/4.972156

[Kod88] Kodak, *Kodak Book of Large-Format Photography*, Rochester NY: Kodak Books, 1988.

[Kod06] Kodak, *Basic Photographic Sensitometry Workbook*, document H-740, Eastman Kodak Company, November 2006.

[Koi96] T. Koizumi, Hwan-Sul Chun and H. Zen, "A new optical detector for a high-speed AF control," in *IEEE Transactions on Consumer Electronics*, vol. 42, no. 4, pp. 1055–1061, Nov 1996. doi: https://doi.org/10.1109/30.555905

[Kok97] C. W. Kok, "Fast algorithm for computing discrete cosine transform," in *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 757–760, Mar 1997. doi: https://doi.org/10.1109/78.558495

[Kok10] E. Kokiopoulou and P. Frossard, "Distributed classification of multiple observations by consensus," in *2010 I.E. International Conference on Image Processing*, Hong Kong, 2010, pp. 2697–2700. doi: https://doi.org/10.1109/ICIP.2010.5652022

[Kom89] T. Komarek and P. Pirsch, "Array architectures for block matching algorithms," in *IEEE Transactions on Circuits and Systems*, vol. 36, no. 10, pp. 1301–1308, Oct 1989. doi: https://doi.org/10.1109/31.44346

[Kon92] Toshiharu Kondo, Akihiro Kikuchi, Takashi Kohashi, Fumiaki Kato, and Katuaki HIrota, "Digital color video camera with auto-focus, auto-exposure, and auto-white balance, and an auto exposure system therefor which compensates for abnormal lighting," U. S. Patent 5,093,716, Mar 3, 1992.

[Kos09] Kosov S., Thormählen T., Seidel HP. (2009) Accurate Real-Time Disparity Estimation with Variational Methods. In: Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2009. Lecture Notes in Computer Science, vol. 5875. Springer, Berlin, Heidelberg

[Kum05] Dinesh Kumar, Pavan Shastry, and Anirban Basu, "Overview of the H.264/AVC," in *8th Texas Instruments Developer Conference India*, Texas Instruments, Nov 30–Dec 1, 2005.

[Kry03] A. I. Krymski, N. E. Bock, N. Tu, D. Van Blerkom, and E. R. Fossum, "A high-speed, 240-frame/s, 4.1-Mpixel CMOS sensor," IEEE Transactions on Electron Devices, 50 (1), January 2003, pp. 130–135.

[Kwa03] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003. "Graphcut textures: image and video synthesis using graph cuts," in *ACM Transactions on Graphics*, vol. 22, no. 3 (July 2003), pp. 277–286. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/882262.882264

[Lan71] E. H. Land and J. J. McCann, "Lightness and retinex theory," in *Journal of the Optical Society of America*, vol. 61, 1971, pp. 1–11.

[Lan72] Edwin H. Land, "Absolute one-step photography," in *Photographic Science and Engineering*, vol. 16, no. 4, July–Aug 1972, pp. 247–257.

[Lar97] G. W. Larson, H. Rushmeier and C. Piatko, "A visibility matching tone reproduction operator for high dynamic range scenes," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 4, pp. 291–306, Oct–Dec 1997. doi: https://doi.org/10.1109/2945.646233

[Lee97] Yung-Pin Lee, Thou-Ho Chen, Liang-Gee Chen, Mei-Juan Chen and Chung-Wei Ku, "A cost-effective architecture for 8×8 two-dimensional DCT/IDCT using direct method," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 459–467, Jun 1997. doi: https://doi.org/10.1109/76.585925

[Lee01] June-Sok Lee, You-Young Jung, Byung-Soo Kim and Sung-Jea Ko, "An advanced video camera system with robust AF, AE, and AWB control," in *IEEE Transactions on Consumer Electronics*, vol. 47, no. 3, pp. 694–699, Aug 2001. doi: https://doi.org/10.1109/30.964165

[Lee03] Lee, Ann B., Kim S. Pedersen, and David Bryant Mumford, "The nonlinear statistics of high-contrast patches in natural images," in *International Journal of Computer Vision*, vol. 54, nos. 1–3, 2003, pp. 83–103.

[Lee06] Sang-Min Lee, Hyunsik Park, and Bruce A. Wooley, "Per-pixel floating-point ADCs with electronic shutters for a high dynamic range, high frame rate infrared focal plane array," in IEEE 2006 Custom Integrated Circuits Conference, IEEE, 2006, pp. 647–650.

[Len87] R. Lenz and R. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3D machine vision metrology," in *Proceedings. 1987 I.E.*

International Conference on Robotics and Automation, 1987, pp. 68–75. doi: https://doi.org/10.1109/ROBOT.1987.1088012

[Len88] R. K. Lenz and R. Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, pp. 713–720, Sept 1988. doi: https://doi.org/10.1109/34.6781

[Leu95] T. K. Leung, M. C. Burl and P. Perona, "Finding faces in cluttered scenes using random labeled graph matching," in *Proceedings of IEEE International Conference on Computer Vision*, Cambridge, MA, 1995, pp. 637-644. doi: https://doi.org/10.1109/ICCV.1995.466878

[Lie02] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings. International Conference on Image Processing*, 2002, pp. I-900-I-903 vol. 1. doi: https://doi.org/10.1109/ICIP.2002.1038171

[Li94] Reoxiang Li, Bing Zeng and M. L. Liou, "A new three-step search algorithm for block motion estimation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 4, pp. 438–442, Aug 1994. doi: https://doi.org/10.1109/76.313138

[Lin06] Douglas O. Linder, "Famous Trials: Oklahoma City Bombing Trial," http://law2.umkc.edu/faculty/projects/ftrials/mcveigh/mcveightrial.html, 2006, accessed Dec 20, 2016.

[Lin10] Chung-Ching Lin and Marilyn Wolf, "Belief Propagation for Detecting Moving Objects from a Moving Platform", in *International Conference on Image Processing, Computer Vision and Pattern Recognition* (IPCV), IEEE, 2010.

[Lin10B] Chang Hong Lin, Marilyn Wolf, Xenefon Koutsoukos, Sandeep Neema, and Janos Sztipanovits. 2010. "System and software architectures of distributed smart cameras," in *ACM Transactions on Embedded Computing Systems*, vol. 9, no. 4, Article 38 (Apr 2010), pp. 30. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/1721695.1721704

[Lin12] Chung-Ching Lin, *Detecting and Tracking Moving Objects From a Moving Platform*, Ph.D. dissertation, Georgia Institute of Technology, April 2012.

[Liu93] B. Liu and A. Zaccarin, "New Fast Algorithms for the Estimation of Block Motion Vectors," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 2, pp. 148–157, Apr. 1993.

[Liu11] T. Liu, Zejian Yuan, Jian Sun, Jindong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum, "Learning to Detect a Salient Object," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 2, pp. 353–367, Feb. 2011. doi: https://doi.org/10.1109/TPAMI.2010.70

[Loe89] C. Loeffler, A. Ligtenberg and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *International Conference on Acoustics, Speech, and Signal Processing*,, Glasgow, 1989, pp. 988-991 vol. 2. doi: https://doi.org/10.1109/ICASSP.1989.266596

[Low99] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, 1999, pp. 1150–1157 vol.2. doi: https://doi.org/10.1109/ICCV.1999.790410

[Low04] David G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, vol. 60, no. 2, 2004, pp. 91–110.

[Low04B] David G. Lowe, *Method and apparatus for identifying scale invariant features in an image and use of same for locating an object in an image*, U. S. Patent 6,711,293, Mar 23, 2004.

[Luc81] Bruce D. Lucas and Takeo Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of Image Understanding Workshop*, 1981, pp. 121–130.

[Lyo02] Richard F. Lyon and Paul M. Hubel, "Eyeing the Camera: Into the Next Century," in *Proc. IS&T/SID 10th Color Imaging Conf.*, 2002, pp. 349–355.

[Mac93] Michael P. MacKay, "Method and apparatus for stabilizing an image produced in a video camera," U. S. Patent 5,253,071, Oct 12, 1993.

[Mal03] H. S. Malvar, A. Hallapuro, M. Karczewicz and L. Kerofsky, "Low-complexity transform and quantization in H.264/AVC," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 598–603, July 2003. doi: https://doi.org/10.1109/TCSVT.2003.814964

[Man95] Steve Mann and Rosiland W. Picard. "Extending dynamic range by combining different exposed pictures," in *Proceedings IS&T Annual Conference*, pp. 442–448, 1995. Also MIT Media Laboratory Perceptual Computing Section Technical Report No. TR-323.

[Mas92] Gerald Mast, *A Short History of the Movies*, fifth edition, revised by Bruce F. Kawin New York: Macmillan Publishing, 1992.

[Mel72] H. Melchior, "Demodulation and Photodetection Techniques," in F. T. Arecchi and E. O. Schulz-Dubois, eds., *Laser Handbook*, vol. 1, Amsterdam: North-Holland, 1972, pp. 725-835.

[Mer10] T. Mertens, J. Kautz, and F. Van Reeth, "Exposure fusion, a simple and practical alternativbe to high dynamic range photography," in *Computer Graphics Forum*, vol. 28, no. 1, March 2009, pp. 161–171, doi: https://doi.org/10.1111/j.1467-8659.2008.01171.x

[Mic00] Microsoft, *Microsoft Extensible Firmware Initiative FAT32 File System Specification*, Version 1.03, Dec 6, 2000.

[Mik05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A Comparison of Affine Region Detectors," in *International Journal of Computer Vision*, vol. 65, no. 1, Nov 1, 2005, pp. 43–72, doi: https://doi.org/10.1007/s11263-005-3848-x

[Mil16] Peyman Milanfar, "Enhance! RAISR sharp images with machine learning," Google Research Blog, Nov 14, 2016, https://research.googleblog.com/2016/11/enhance-raisr-sharp-images-with-machine.html

[Nak05] Junichi Nakamura, *ed*., *Image Sensors and Signal Processing for Digital Still Cameras*, CRC Press, 2005.

[Nas08] H. H. Nasse, "How to Read MTF Curves," Carl Zeiss, Camera Lens Division, Dec 2008.

[Net79] A. N. Netravali and J. D. Robbins, "Motion-compensated television coding: Part I," in *The Bell System Technical Journal*, vol. 58, no. 3, pp. 631–670, Mar 1979. doi: https://doi.org/10.1002/j.1538-7305.1979.tb02238.x

[New64] Beaumont Newhall, *The History of Photography*, revised and enlarged edition, New York: The Museum of Modern Art, 1964.

[Nie85] Niels J. Nielsen, "History of ThinkJet Printhead Development," in *Hewlett-Packard Journal*, vol. 36, no. 5, May 1985, pp. 4–10.

[Nix96] R. H. Nixon, S. E. Kemeny, B. Pain, C. O. Staller and E. R. Fossum, "256×256 CMOS active pixel sensor camera-on-a-chip," in *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, pp. 2046–2050, Dec 1996. doi: https://doi.org/10.1109/4.545830

[NVI14] NVIDIA, *NVIDIA GeForce GTX 980*, white paper, 2014.

[NVI14B] NVIDIA, *NVIDIA Jetson TX1 System-on-Module*, data sheet, 2014.

[Oh04] Songhwai Oh, Stuart Russell, and Shankar Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems, in *43rd IEEE Conference on Decision and Control*, IEEE, 2004, TuB08.5

[Oiz93] Kouji Oizumi, Nozomu Kitagishi, and Shoichi Yamzaki, "Optical system for stabilizing an image," U. S. Patent 5,270,857, Dec 14, 1993.

[Pal99] Stephen E. Palmer, *Vision Science: Photons to Phenomenology,* Cambridge MA: MIT Press, 1999.

[Par62] Parzen, Emanuel. "On Estimation of a Probability Density Function and Mode." in *The Annals of Mathematical Statistics*, vol. 33, no. 3, 1962, pp. 1065–1076. www.jstor.org/stable/2237880.

[Par97] Kenneth A. Parulski and James E. McGarvey, "Automatic camera exposure control using variable exposure index CCD sensor," U. S. Patent 5,610,654, March 11, 1997.

[Pat97] A. J. Patti, M. I. Sezan and A. Murat Tekalp, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," in *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1064–1076, Aug 1997. doi: https://doi.org/10.1109/83.605404

[Per07] Michael R. Peres, ed., *The Focal Encyclopedia of Photography: Digital Imaging, Theory and Applications, History, and Science,* Burlington MA: Focal Press, 2007.

[Phi07] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 I.E. Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007, pp. 1–8. doi: https://doi.org/10.1109/CVPR.2007.383172

[Po96] Lai-Man Po and Wing-Chung Ma, "A novel four-step search algorithm for fast block motion estimation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313–317, Jun 1996.

[Pop14] Vladan Popovic, Kerem Seyid, Elieva Pignat, Omer Cogal, Yusuf Leblebici, "Multi-camera platform for panoramic real-time HDR video construction and rendering," in *Journal of Real-Time Image Processing*, vol. 26 July 2014, doi: https://doi.org/10.1007/s11554-014-0444-8

[Rei79] D. Reid, "An algorithm for tracking multiple targets," in *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec 1979. doi: https://doi.org/10.1109/TAC.1979.1102177

[Rei93] R. K. Reich *et al*., "Integrated electronic shutter for back-illuminated charge-coupled devices," in *IEEE Transactions on Electron Devices*, vol. 40, no. 7, pp. 1231–1237, Jul 1993. doi: https://doi.org/10.1109/16.216426

[Rin08] B. Rinner and W. Wolf, "An Introduction to Distributed Smart Cameras," in *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1565–1575, Oct 2008. doi: https://doi.org/10.1109/JPROC.2008.928742

[Rob10] M. D. Robinson, C. A. Toth, J. Y. Lo and S. Farsiu, "Efficient Fourier-Wavelet Super-Resolution," in *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2669–2681, Oct 2010. doi: https://doi.org/10.1109/TIP.2010.2050107

[San17] Rishi Sanyal, "Sony FE 100mm F2.8 STF bokeh demystified," dpreview.com, Feb 17, 2017.

[Say15] Rishi Saynal, "Sony Alpha 7R II: real-world ISO invariance study," *Digital Photography Review*, Aug 24, 2015, https://www.dpreview.com/articles/7450523388/sony-alpha-7r-ii-real-world-iso-invariance-study

[Sch98] Schneider Kreuznach, "Optics for Digital Photography: A White Paper," Schneider Kreuznach, 1998.

[Seq75] Carlo H. Se´quin and Michael F. Tompsett, *Charge Transfer Devices*, New York: Academic Press, 1975.

[Sch64] O. H. Schade, "An Evaluation of Photographic Image Quality and Resolving Power," in *Journal of the SMPTE*, vol. 73, no. 2, pp. 81–119, Feb 1964. doi: https://doi.org/10.5594/J06117

[Sch00] M. Schanz, C. Nitta, A. Bussmann, B. J. Hosticka and R. K. Wertheimer, "A high-dynamic-range CMOS image sensor for automotive applications," in *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 932–938, July 2000. doi: https://doi.org/10.1109/4.848200

[Sch07] J. Schlessman, M. Lodato, B. Ozer and W. Wolf, "Heterogeneous MPSoC Architectures for Embedded Computer Vision," in *2007 I.E. International Conference on Multimedia and Expo*, Beijing, 2007, pp. 1870–1873. doi: https://doi.org/10.1109/ICME.2007.4285039

[Sch15] Jason Schlessman and Marilyn Wolf, "Tailoring design for embedded computer vision applications," in *IEEE Computer,* vol. 48, no. 5, May 2015, pp. 58–62.

[Sha94] S. Shah and J. K. Aggarwal, "A simple calibration procedure for fish-eye (high distortion) lens camera," in *Proceedings of the 1994 I.E. International Conference on Robotics and Automation*, San Diego, CA, 1994, pp. 3422–3427 vol.4. doi: https://doi.org/10.1109/ROBOT.1994.351044

[She05] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1778–1792, Nov 2005. doi: https://doi.org/10.1109/TPAMI.2005.213

[Siv03] J. Sivic and A. Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1470–1477, vol. 2. doi: https://doi.org/10.1109/ICCV.2003.1238663

[Spr12] Kenneth R. Spring, John C. Russ, Matthew Parry-Hill, Thomas J. Fellers, and Michael W. Davidson, "Unsharp mask mkiltering," 2012, http://olympus.magnet.fsu.edu/primer/java/digitalimaging/processing/unsharpmask/index.html

[Sta76] Norman L. Stauffer, "Auto-focus camera with solid state range finder," U. S. Patent 3.945,023, Mar 16, 1976.

[Sta99] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings. 1999 I.E. Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Fort Collins, CO, 1999, pp. 252, vol. 2. doi: https://doi.org/10.1109/CVPR.1999.784637

[Sta00] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, Aug 2000. doi: https://doi.org/10.1109/34.868677

[Sto02] D. Stoppa, A. Simoni, L. Gonzo, M. Gottardi and G. F. Dalla Betta, "Novel CMOS image sensor with a 132-dB dynamic range," in *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1846–1852, Dec 2002. doi: https://doi.org/10.1109/JSSC.2002.804347

[Sul91] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," *Global Telecommunications Conference, 1991. GLOBECOM '91. 'Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services*, Phoenix, AZ, 1991, pp. 85–90 vol.1. doi: https://doi.org/10.1109/GLOCOM.1991.188361

[Sun03] Jian Sun, Nan-Ning Zheng, Hai Tao and Heung-Yeung Shum, "Image hallucination with primal sketch priors," in *2003 I.E. Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2003, pp. II-729-36 vol. 2. doi: https://doi.org/10.1109/CVPR.2003.1211539

[Sze81] S. M. Sze, *Physics of Semiconductor Devices*, second edition, New York: John Wiley and Sons, 1981.

[Sze97] Richard Szeliski and Heung-Yeung Shum. 1997. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '97). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 251–258. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/258734.258861

[Tak02] Hiroaki Takada, editor, and Ken Sakamura, supervisor, *ITRON4.0 Specification,* version 4.00.00, Tron Association, 2002.

[Tex16] Texas Instruments, *AM572x Sitara™ Processors Silicon Revision 2.0, 1.1 Technical Reference Manual*, Literature Number SPRUHZ6H, Oct 2014, revised Nov 2016.

[Tha98] Jo Yew Tham, S. Ranganath, M. Ranganath and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 369–377, Aug 1998. doi: https://doi.org/10.1109/76.709403

[The04] T. Theocharides, G. Link, N. Vijaykrishnan, M. J. Irwin and W. Wolf,. "Embedded hardware face detection," in *17th International Conference on VLSI Design. Proceedings.*, 2004, pp. 133–138. doi: https://doi.org/10.1109/ICVD.2004.1260915

[The07] A. Theuwissen, "CMOS image sensors: State-of-the-art and future perspectives," in *ESSCIRC 2007 – 33rd European Solid-State Circuits Conference*, Munich, 2007, pp. 21–27. doi: https://doi.org/10.1109/ESSCIRC.2007.4430242

[Tia99] Hui Tian, Boyd A. Fowler, Abbas El Gamal; Analysis of temporal noise in CMOS APS. In *Proc. SPIE 3649*, Sensors, Cameras, and Systems for Scientific/Industrial Applications, 177 (Apr 27, 1999); doi:https://doi.org/10.1117/12.347073.

[Tia00] Hui Tian, Abbas El Gamal, Analysis of 1/f noise in CMOS APS. In *Proc. SPIE 3965*, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications, 168 (May 15, 2000); doi:https://doi.org/10.1117/12.385433.

[Tit11] Albert H. Titus, Maurice C.-K. Cheung, and Vamsy P. Chodavarapu, "CMOS Photodetectors," Chapter 4 in Jeong-Woo Park, *ed*., Photodiodes/Book 2**,** July 2011.

[Tom08] F. Tombari, S. Mattoccia, L. Di Stefano and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *2008 I.E. Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1–8. doi: https://doi.org/10.1109/CVPR.2008.4587677

[Tom91] Carlo Tomasi and Takeo Kanade, *Detection and Tracking of Point Features*, Technical Report CMU-CS-91-132, Carnegie Mellon University, Apr 1991.

[Tru83] Francois Truffaut, *Hitchcock, Revised Edition*, New York: Simon and Schuster, 1983.

[Tru85] Francois Truffaut and Helen Scott, Hitchcock, revised edition, New York: Simon and Schuster, 1985.

[Tsa84] R. Tsai and T. Huang, "Multiframe image restoration and registration," in *Advances in Computer Vision and Image Processing*, vol. 1, Greenwich, CT: JAI, 1984.

[Tsa87] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," in *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, Aug 1987. doi: https://doi.org/10.1109/JRA.1987.1087109

[Tsu08] Hidetoshi Tsubaki, Mitsuhiro Saito, and Takahiro Oshino, "Image stabilizing apparatus, image-pickup apparatus and image stabilizing method," U. S. Patent Application Publication 2008/0246848 A1, Oct 9, 2008.

[Tsu93] Akio Tsuji and Teruo Sano, "Electronic camera with automatic exposure control," U. S. Patent 5,223,935, June 29, 1993.

[Tto15] Christos Ttofis, Christos Kyrkou, and Theocharis Theocharides. 2015. "A Hardware-Efficient Architecture for Accurate Real-Time Disparity Map Estimation," in *ACM Transactions on Embedded Computing Systems,* vol. 14, no. 2, Article 36 (Feb 2015), pp. 26. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/2629699

[Tur91] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Proceedings. 1991 I.E. Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, 1991, pp. 586–591. doi: https://doi.org/10.1109/CVPR.1991.139758

[Unk63] Unknown photographer, "Unidentified young African American soldier in Union uniform," taken between 1863 and 1865, hand-colored tintype, Reproduction Number LC-DIG-ppmsca-50221, Call Number AMB/TIN no. 3264, Library of Congress Prints and Photographs Division Washington, D.C. 20540 USA http://hdl.loc.gov/loc.pnp/pp.print

[Unk03] Unknown photographer, "Original Wright Brothers' 1903 Aeroplane ("Kitty Hawk") in first flight, Dec 17, 1903 at Kitty Hawk, N.C. Orville Wright at controls. Wilbur observing." National Archives Identifier: 7580929 Local Identifier: 165-WW-713-6. Creator: War Department. 1789-9/18/1947

[Vel05] S. Velipasalar and W. Wolf, "Multiple object tracking and occlusion handling by information exchange between uncalibrated cameras," in *IEEE International Conference on Image Processing 2005*, 2005, pp. II-418–21. doi: https://doi.org/10.1109/ICIP.2005.1530081

[Vel06] S. Velipasalar, J. Schlessman, C. Y. Chen, W. Wolf and J. P. Singh, "SCCS: A Scalable Clustered Camera System for Multiple Object Tracking Communicating Via Message Passing Interface," in *2006 I.E. International Conference on Multimedia and Expo*, Toronto, Ont., 2006, pp. 277–280. doi: https://doi.org/10.1109/ICME.2006.262452

[Vel08] Senem Velipasalar and Wayne H. Wolf, "Frame-level temporal calibration of video sequences from unsynchronized cameras," *Machine Vision and Applications Journal*, Jan 2008, doi: https://doi.org/10.1007/s00138-008-0122-6

[Vio01] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 I.E. Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, pp. I-511-I-518 vol.1. doi: https://doi.org/10.1109/CVPR.2001.990517

[Wal91] Gregory K. Wallace. 1991. "The JPEG still picture compression standard," in *Communications of the ACM*, vol. 34, no. 4 (Apr 1991), pp. 30–44. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/103085.103089

[Wan04] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in *IEEE Transactions on Image Processing,* vol. 13, no. 4, Apr 2004, pp. 600–612.

[Wan16] Wei-Chih Wang, "Optical Detectors," undated notes.

[Whi74] M. H. White, D. R. Lampe, F. C. Blaha and I. A. Mack, "Characterization of surface channel CCD image arrays at low light levels," in *IEEE Journal of Solid-State Circuits*, vol. 9, no. 1, pp. 1–12, Feb 1974. doi: https://doi.org/10.1109/JSSC.1974.1050448

[Whi11] Nathan Whitehead and Alex Fit-Florea, "Precision & performance: floating point and IEEE 754 compliance for NVIDIA GPUs," NVIDIA, 2011.

[Wid08] W. H. Widen, "Smart Cameras and the Right to Privacy," in *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1688–1697, Oct 2008. doi: https://doi.org/10.1109/JPROC.2008.928764

[Wie03] T. Wiegand, G. J. Sullivan, G. Bjontegaard and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003. doi: https://doi.org/10.1109/TCSVT.2003.815165

[Wie03B] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003. doi: https://doi.org/10.1109/TCSVT.2003.815168

[Wol96] Wayne Wolf, "Key frame selection by motion analysis," in *Proceedings, ICASSP '96*, IEEE Press, 1996, pp. 1240–1243.

[Wol02] Wayne Wolf, Burak Ozer, and Tiehan Lv, "Smart cameras as embedded systems," in *IEEE Computer*, vol. 35, no. 9, Sept 2002, pp. 48–53.

[Wol08] Wayne Wolf, Ahmed A. Jerraya, and Grant Martin, "Multiprocessor System-on-Chip (MPSoC) Technology," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 27, no. 10, Oct 2008, pp. 1701–1713.

[Wol17] Marilyn Wolf, *The Physics of Computing*, Cambridge MA: Elsevier, 2017.

[Wre97] C. R. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, "Pfinder: real-time tracking of the human body," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, Jul 1997. doi: https://doi.org/10.1109/34.598236

[Xu12] R. Xu, B. Liu and J. Yuan, "A 1500 fps Highly Sensitive 256 X 256 CMOS Imaging Sensor With In-Pixel Calibration," in *IEEE Journal of Solid-State Circuits*, vol. 47, no. 6, pp. 1408–1418, June 2012. doi: https://doi.org/10.1109/JSSC.2012.2192662

[Yad97] Orly Yadid-Pecht, Karmak Mansoorian, Eric R. Fossum, Bedabrata Pain; Optimization of noise and responsivity in CMOS active pixel sensors for detection of ultralow-light levels. Proc. SPIE 3019, Solid State Sensor Arrays: Development and Applications, 125 (Apr 25, 1997); doi: https://doi.org/10.1117/12.275185.

[Yam81] Akira Yamanaka and Toshinori Imura, "Auto-focus camera having a rangefinder," U. S. Patent 4,300,823, Nov 17, 1981.

[Yam06] K. Yamaguchi, T. Kato and Y. Ninomiya, "Vehicle Ego-Motion Estimation and Moving Object Detection using a Monocular Camera," in *18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong, 2006, pp. 610–613. doi: https://doi.org/10.1109/ICPR.2006.1165

[Yan89] K. M. Yang, M. T. Sun and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm," in *IEEE Transactions on Circuits and Systems*, vol. 36, no. 10, pp. 1317–1325, Oct 1989. doi: https://doi.org/10.1109/31.44348

[Yan02] Ming-Hsuan Yang, D. J. Kriegman and N. Ahuja, "Detecting faces in images: a survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan 2002. doi: https://doi.org/10.1109/34.982883

[Yan05] Shengqi Yang, Wayne Wolf and Narayanan Vijaykrishnan, "Power and performance analysis of motion estimation based on hardware and software realizations," in *IEEE Transactions on Computers*, vol. 54, no. 6, June 2005, pp. 714–726.

[Yan10] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," IEEE Transactions on Image Processing, 19(11), November 2010, pp. 2861–2873.

[Zab94] Ramin Zabih and John Woodfill, "Non-parametric local transforms for computing visual correspondence," in Jan-Olof Eklundh, ed., *Computer Vision --- ECCV '94: Third European Conference on Computer Vision Stockholm, Sweden, May 2–6 1994 Proceedings, Volume II*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 151–158, doi https://doi.org/10.1007/BFb0028345

[Zaj05] Wojciech Zajdel and Ben J. A. Kröse, "A sequential Bayesian algorithm for surveillance with nonoverlapping cameras," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, pp. 977 (2005). doi: https://doi.org/10.1142/S0218001405004423

[Zha03] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. 2003. "Face recognition: A literature survey," in *ACM Computing Surveys,* vol, 35, no. 4 (Dec 2003), pp. 399–458. doi: http://dx.doi.org.prx.library.gatech.edu/10.1145/954339.954342

[Zhe11] R. Zheng, T. Wei, D. Gao, Y. Zheng, F. Li and H. Zeng, "Temporal noise analysis and optimizing techniques for 4-T pinned photodiode active pixel sensor," in *2011 I.E. International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Xi'an, 2011, pp. 1–5. doi: https://doi.org/10.1109/ICSPCC.2011.6061804

[Zhe13] W. S. Zheng, S. Gong and T. Xiang, "Reidentification by Relative Distance Comparison," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 653–668, Mar 2013 doi: https://doi.org/10.1109/TPAMI.2012.138

[Zho04] S.K. Zhou, R. Chellappa, and B. Moghaddam. "Visual tracking and recognition using appearance-adaptive models for particle filters," in *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[Zho12] D. Zhou, X. Shen and W. Dong, "Image zooming using directional cubic convolution interpolation," in *IET Image Processing*, vol. 6, no. 6, pp. 627–634, Aug 2012. doi: https://doi.org/10.1049/iet-ipr.2011.0534

[Zhu00] Ying Zhu, S. Schwartz and M. Orchard, "Fast face detection using subspace discriminant wavelet features," *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, Hilton Head Island, SC, 2000, pp. 636–642 vol. 1. doi: https://doi.org/10.1109/CVPR.2000.855879

[Zon11] M. Zontak and M. Irani, "Internal statistics of a single natural image," *CVPR 2011*, Providence, RI, 2011, pp. 977–984. doi: https://doi.org/10.1109/CVPR.2011.5995401

[Zuc16] Esther Zuckerman, "FX counts a whopping 455 original scripted series this year," avclub.com, Dec 21, 2016.