Chapter 7

Vision

The applications of digitial image processing have grown tremendously in the past decade. The role for processing data sets naturally represented as two dimensional data structures is growing still. Remote sensing applications are used to work in hostile environments, to reconstruct the surface topography of the earth from high altitude fly-overs, and to image various physical phenomena inside of the human body.

This chapter focuses specifically on techniques for *closing the loop* in robotic mechanisms. Biological systems call on a variety of percepts to construct a richly encoded representation of the relationship between the organism and the world. Humans use vestibular, cutaneous tactile, visual, auditory, olfactory, and proprioceptive feedback in this regard. In this chapter, we will consider visual feedback by examining techniques derived from computer vision in some detail. Many of the topics discussed are generic and fall under the rubric of signal processing. In this sense they apply to all perceptual tasks that involve the interpretation of a temporal signal.

However, in this chapter we will discuss a more specific question — how might image processing be used to direct the behavior of a robotic system? Traditionally, the robotics community has adopted a viewpoint we will refer to as *percept inversion*. This approach advocates asking the following question; if sensory stimuli are produced in such and such a manner, then what must the world have been like to produce this stimulus? Stated another way, if stimulus can be predicted given assumptions regarding the world,

$$Stimulus = f(World)$$

then the world can be reconstructed from a pattern of stimulation

$$World = f^{-1}(S).$$

The trouble with this perspective is that often the function $f()$ is only partially known, and in general, the inverse of $f()$ is not well-conditioned.

Many researchers in artificial intelligence and robotics are now considering techniques designed to handle precisely those situations when the relevant world state in not instantaneously accessible. These approaches attempt to use knowledge and experience in the form of a time series of observable state information to fill in initially inaccessible detail. This approach can require many actions on the part of the robot to generate the *right* time series with which to determine critical state variables so the approach is inherently active. For this reason, one thinks of such a robot as a embodied perceptual system.

An example of the perspective offered by the two paradigms as they pertain to computer vision tasks was presented by Bob Bolles at the 1993 International Symposium on Intelligent Control.

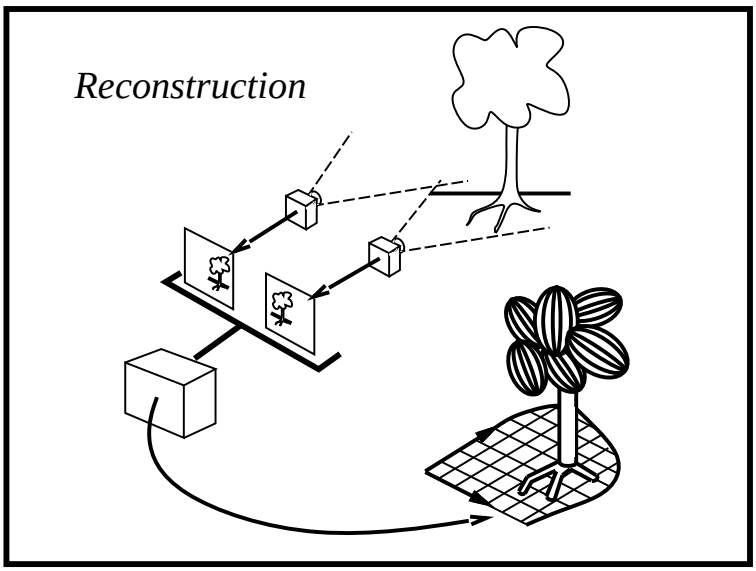

Figure 7.1 *Static Reconstruction Architecture and Task.*

In Figure 7.1 a stereo pair is used to reconstruct the world geometry. This task requires specialized architectures and algorithms and is often the approach employed to interpret medical imagery or to construct topological maps from high altitude fly overs.

However, another kind of task can be specified. Suppose that a mobile robot must navigate across outdoor terrain and must avoid intervening obstacles.

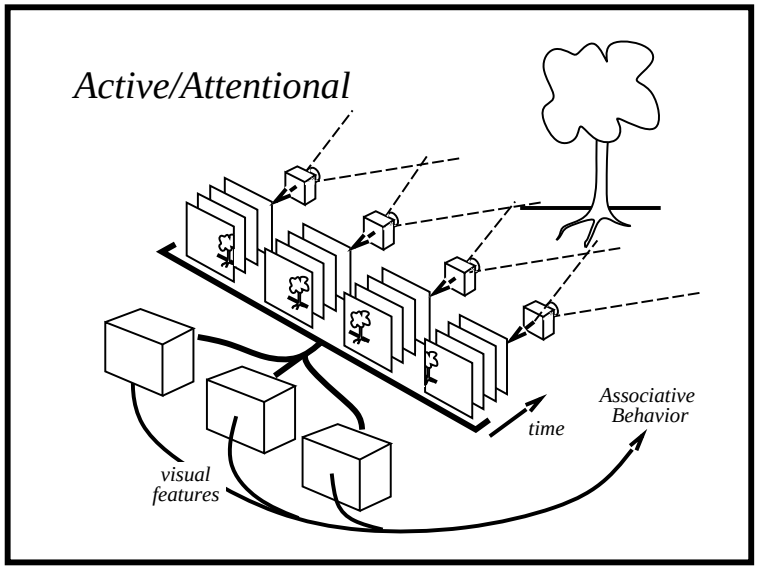

Figure 7.2 *Dynamic Attentional Architecture.*

Many of the details provided by reconstruction techniques are likely to be irrelevant in this task and the responsiveness of the robot to the obstacle may well depend on how precisely the robot focuses on just the right visual feature set. In other words, a looming feature may be just the level of detail for the navigation task.

Although much of what is discussed in this chapter is generally applicable to arbitrary vision tasks, the emphasis of these notes is on tasks like the navigation task where less interpretation is required.

7.1 Introduction

Vision in biology and in machine is fundamentally an indirect source of information. The sensor itself responds to incident electromagenetic energy, but the *reason* that we look is to ascertain geometric properties of the world around us. This information is very subtly encoded in the patterns of electromagnetic energy falling on the 2 dimensional imaging plane.

Figure 7.3 illustrates a sequence of transformations between the illumination source and a digital image. Energy from the illumination source is radiated uniformly over 4π steradians, its radiant intensity attentuated as the inverse square of the distance from the source. The total amount of light energy falling on a surface is referred to as *irradiance*[1] and consists of the sum of all incident

[1] On the surface of the earth, the sun projects about 1200 $Watts/m^2$ (or 429 $BTUH/ft^2$).

energy from all sources. The total energy leaving the surface is referred to as its *radiance* and differs from the irradiance by energy transmitted into and absorbed by the material. As shown in the diagram, sometimes objects nearby act as indirect sources by reflecting energy toward other objects. In doing so, these secondary sources modulate the spectral content of the original source, and they may more rapidly attenuate the light intensity through diffuse reflection and polarization. In order to observe a portion of the radiant intensity function, the 3 dimensional world is projected onto a 2 dimensional image plane that is sampled at regular intervals, digitized with finite resolution, and written into a digital memory.

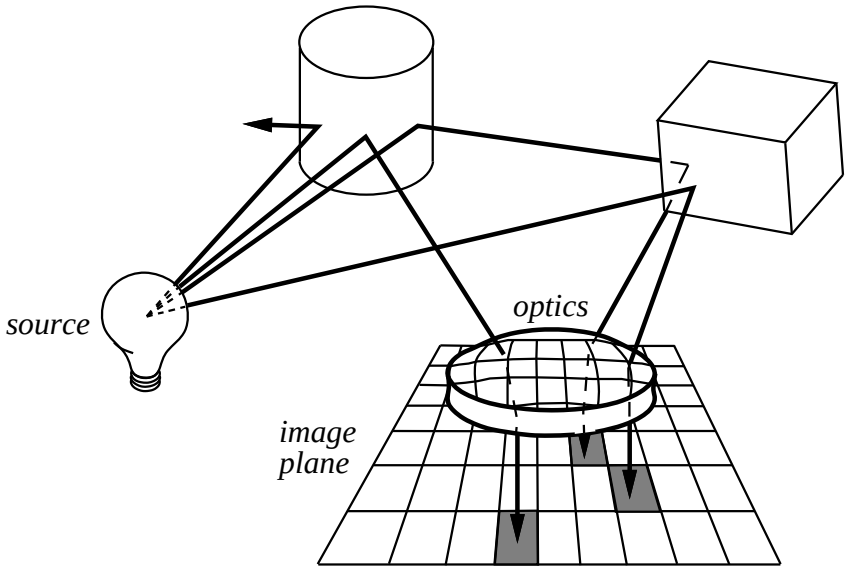

Figure 7.3 *The image formation transformations.*

Our goal is to extract information about the world from a 2D projection of the energy stream derived from this terribly complex 3D interaction with the world. In the next several sections we will introduce machine vision and formulate models for projective image geometry. We will examine the effects of sampling and digitization and we will discuss techniques for identifying features on the image plane that predict relevant world state information. We will introduce methods for identifying conjunctive feature sets in situations where spatial constraints between features are known. Finally, in order to relate vision to motor control, we will illustrate how sequences of images can be used to close the sensorimotor loop, and how information can be fused over time and over sensor modality.

7.2 Human Eye

Figure 7.4 is a sketch of the anatomy of the human eyeball. Light enters the eye through the transparent cornea and is focused by a lens that changes shape under muscular control. The iris acts as a shutter to control the amount of light entering the eye.

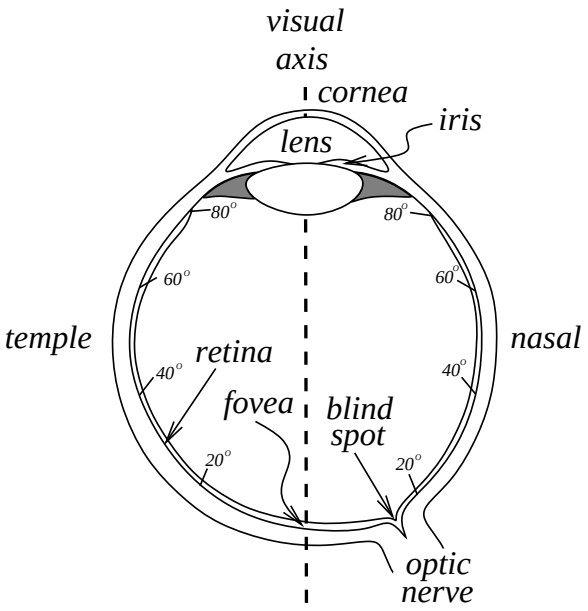

Figure 7.4 *A cross section of the human eyeball.*

The photosensitive surface of the eye is called the retina consisting of rod and cone receptors. The rods are more sensitive to incident light and are about twice as numerous as the cones. The cones, however, are specialized receptors responding to red, green and blue wavelengths within the visual spectrum. When stimulated, these receptors produce impulses in the retinal cells. The greatest concentration of both receptors is near the fovea, however, this is by far the greatest concentration of cones anywhere on the retina. There are about 100×10^6 receptors and roughly 0.8×10^6 nerve fibers exiting in the optic nerve. This suggests that certain forms of local signal processing occurs directly on the retina.

Later in this chapter, we will describe the manner in which receptors on the image plane sample the image function (Section 7.4) and we will introduce image operations that require only local neighborhoods of the retina (Section 7.5).

7.3 Imaging Geometry

7.3.1 The Pinhole Camera

Figure 7.5 illustrates the classical pinhole camera geometry with which we can model the perspective projection. Figure 7.5(a) depicts an imaging geometry that describes the projective transformation in the eye and in machine vision systems. The imaging surface is exposed to the electromagnetic radiosity function through a small aperture so that radiant intensities on the image plane correspond geometrically to features of the 3D world. Through a simple similar triangles construction, we see that the u coordinate on the image plane if proportional to the x coordinate of the 3D feature and

inversely proportional to its 3D range, z.

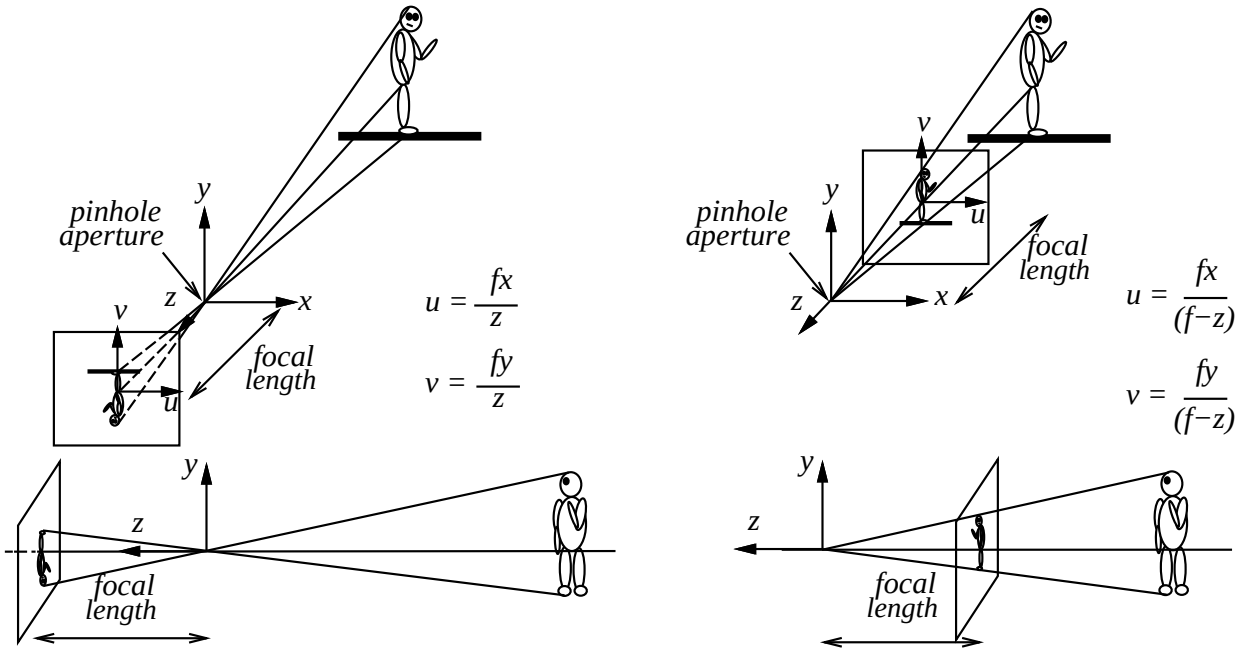

Figure 7.5 *Pinhole camera model: (a) perspective projection geometry. (b) mathematically equivalent non-inverting geometry.*

Note that the image is inverted by this projection, as is the case in the human eye. The brain adapts to this projection, employing visual and vestibular information to register geometry and force sensations. Figure 7.5(b) presents a mathematically equivalent projective geometry that does not invert the image.

7.3.2 Gaussian Lens Formula

Pinhole cameras have an infinite depth of field - that is, they focus information onto the image plane from faraway as well as nearby features. The geometry of the projection is determined by the pinhole. This is mathematically sound (and actually works), but the pinhole permits only very small amounts of electromagnetic information onto the focal plane and can do little to control the field of view. To collect more light and to control field of view, optics are placed between the scene and the imaging plane. As a consequence, we discard the infinite depth of field that was so useful in the pinhole camera. To manipulate an electromagnetic wavefront before it gets to the image plane one may introduce reflective or refractive elements.

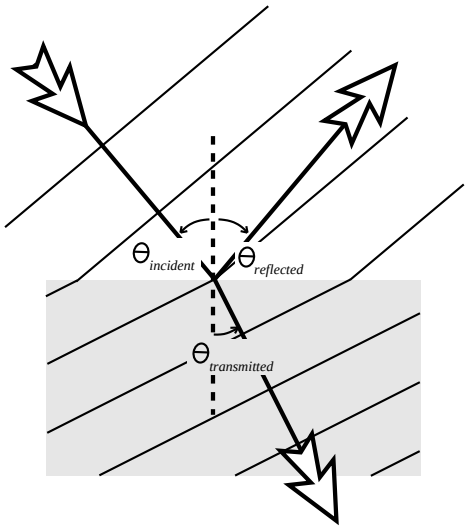

Figure 7.6 *Refraction at an optical interface.*

The most common device used in this manner is the *lens* - in fact, biological systems use deformable lens in a myriad of vision acquisition tasks.

The index of refraction for a material that transmits electromagnetic information is the ratio of the speed of light in a vacuum to that in the optical material.

$$n = \frac{c}{v} = \sqrt{\frac{\epsilon\mu}{\epsilon_0\mu_0}}$$

where μ is the magnetic permeability of the material (μ_0 is the permeability free space) and ϵ is known as the electric permittivity (ϵ_0 is the permittivity of free space).

As a light ray moving in air, for example, crosses an optical interface into material whose index of refraction is greater than that of air, the light rays is refracted, bending the wavefront toward the normal of the optical interface. The phenomenon is described by Snell's law

$$\frac{sin(\theta_{incident})}{sin(\theta_{transmitted})} = \frac{n_t}{n_i}, \tag{7.1}$$

where n_i is the index of refraction for the medium through which the incident light ray travels, and n_t is the index of refraction for the medium through which the transmitted light ray travels.

For typical optical systems, the position of the focal plane with respect to the optics is controllable and we actively control focus. This process brings elements of the scene into focus on the image plane as a function of range. The Gaussian lens formula shows how this works.

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{f} \tag{7.2}$$

where:

Z – distance from the lens to the object

Z' – distance from the lens to place where the image is formed

f – focal length of the lens

Equation 7.2 is a special case of the so-called *lensmaker's formula* written for thin lenses.

Figure 7.7 shows how this works. Light from distant objects (parallel rays on the left) come to focus at f the focal length of the lens. As the object approaches the camera, the light reaching

the lens becomes increasingly divergent, and the point at which it is brought into focus moves correspondingly further from the lens.

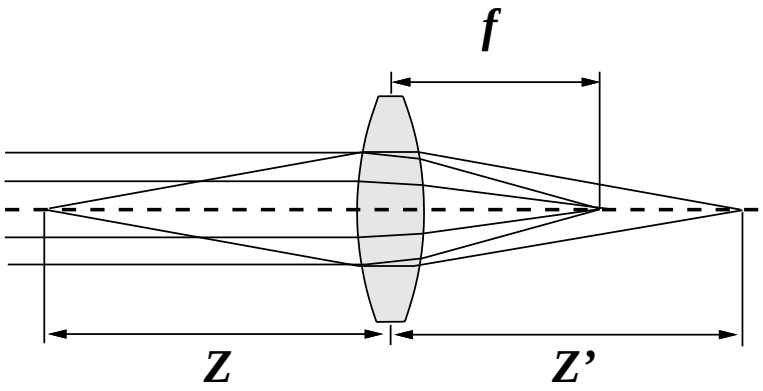

Figure 7.7 *The effects of range, Z, on the distance to the focal plane.*

7.4 Spatial properties of the image function

Both the retina and machine vision equivalents sample the continuous image function. It is clear that such a discrete sampling will discard some of the latent information, what may not be initially clear, is that this sampling may also introduce information not originally in the continuous function. In this section, we will characterize the sampling operator mathematically, discuss its influence on the spectral content of the signal, and introduce the notion of aliasing on the image plane.

A mathematical abstraction that is central to the analysis is that of the Dirac delta function. This operator is referred to as a singularity operator for reasons that are obvious by its definition:

$$\delta(x-\xi, y-\eta) = \begin{cases} \infty & x=\xi, y=\eta \\ 0 & otherwise \end{cases} \tag{7.3}$$

This function has the following properties:

$$\iint_{-\epsilon}^{\epsilon} \delta(x,y)dxdy = 1 \quad for\ \epsilon > 0$$

$$\iint_{-\infty}^{\infty} F(\xi,\eta)\delta(x-\xi, y-\eta)d\xi d\eta = F(x,y)$$

the first of these properties suggests a kind of infinitesimal mask that samples the image function precisely at position (x, y), and the second is referred to as the Sifting property.

7.4.1 Fourier Transform

The Fourier transform of a one dimensional function $f(x)$, is defined as:

$$\mathcal{F}[f(x)] = F(u) = \int_{-\infty}^{\infty} f(x) exp\{-i(2\pi ux)\} dx \tag{7.4}$$

where u is the spatial frequency (in *cycles/pixel* perhaps, so that when x is specified in pixels, $(2\pi ux)$ is in radians, and $i = \sqrt{-1}$. One may view the Fourier transform as the projection of the image function, $f(x)$, onto the basis functions, $exp\{-i(2\pi ux)\}$, for a particular combination of spatial frequencies $u \in [-\infty, \infty]$. The inverse transform is written:

$$\mathcal{F}^{-1}[F(u)] = f(x) = \int_{-\infty}^{\infty} F(u) exp\{i(2\pi ux)\} du \tag{7.5}$$

The corresponding definitions in 2D are:

$$\mathcal{F}[f(x,y)] = F(u,v) = \int_{-\infty}^{\infty}\int f(x,y) exp\{-i(2\pi(ux+vy)\} dxdy \tag{7.6}$$

$$\mathcal{F}^{-1}[F(u,v)] = f(x,y) = \int_{-\infty}^{\infty}\int F(u,v) exp\{i(2\pi(ux+vy))\} dudv \tag{7.7}$$

As we shall see in the following sections, transforming the spatial properties of the image function into the frequency domain provides the basis for a very general spatial frequency filter. Table 7.1 summarizes some of the Fourier transform pairs that we will need for subsequent discussions.

7.4.2 Shift and Convolution Theorems

The Shift theorem follows directly from the definition of the Fourier transform. If

$$\begin{aligned}
\mathcal{F}[f(x)] &= \int_{-\infty}^{\infty} f(x) exp\{-i(2\pi ux)\} dx, \ then \\
\mathcal{F}[f(x-a)] &= \int_{-\infty}^{\infty} f(x-a) exp\{-i(2\pi ux)\} dx, \ then \\
&= \int_{-\infty}^{\infty} f(x') exp\{-i(2\pi u(x'+a))\} dx', \ and \\
&= exp\{-i(2\pi ua)\} \int_{-\infty}^{\infty} f(x') exp\{-i(2\pi ux')\} dx', \ so \ that, \\
\mathcal{F}[f(x-a)] &= exp\{-i(2\pi ua)\} \mathcal{F}(f(x))
\end{aligned} \tag{7.8}$$

Table 7.1: Fourier Transform Pairs

$$F(\omega) = \mathcal{F}(f(x)) = \int_{-\infty}^{\infty} f(x)e^{-i\omega x}dx \qquad \omega(rad/pixel) = 2\pi u(cycles/pixel)$$

Name	$f(x)$	$F(\omega)$
rectangular function	$rect(x) = 1 \quad -\frac{1}{2} < x < \frac{1}{2}$	$sinc(\omega/2\pi) = \frac{sin(\omega/2)}{\omega/2}$
triangular function	$tri(x) = 2(x+\frac{1}{2}) \quad -\frac{1}{2} < x < 0$ $1 - 2(x) \quad 0 < x < \frac{1}{2}$	$sinc^2(\omega/2\pi)$
Gaussian	$e^{-\alpha\|x\|}$ e^{-px^2}	$2\alpha/(\alpha^2+\omega^2)$ $\frac{1}{\sqrt{2p}}e^{-\omega^2/4p}$
unit impulse	$\delta(x)$	1
comb function	$\sum_n \delta(x - nx_0)$	$\frac{1}{x_0}\sum_n \delta(\frac{\omega}{2\pi} - \frac{n}{x_0})$
differentiation	$g^n(x)$	$(i\omega)^n G(\omega)$
linear combination	$ag(x) + bh(x)$	$aG(\omega) + bH(\omega)$
scale	$f(ax)$	$\frac{1}{\|a\|}F(\frac{\omega}{a})$

The convolution of two functions $f(x)$ and $g(x)$, written $f(x) * g(x)$, is defined by the integral,

$$h(x) = \int_{-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha \tag{7.9}$$

where α an integration variable. We will show later that if $f()$ is a so-called convolution operator, then $h(x)$ is equivalent to the pixel-wise correlation of $f()$ and its *footprint* around $g(x)$. An important property of convolution lies in the way in which it maps through the Fourier transform.

$$\begin{aligned}
\mathcal{F}[f(x) * g(x)] &= \mathcal{F}[h(x)] \\
&= \mathcal{F}\left[\int_\alpha f(\alpha)g(x-\alpha)d\alpha\right] \\
&= \int_x \left[\int_\alpha f(\alpha)g(x-\alpha)d\alpha\right] exp\{-i2\pi ux\}dx \\
&= \int_\alpha f(\alpha)\left[\int_x g(x-\alpha)exp\{-i(2\pi ux)\}dx\right] d\alpha \text{ and by the Shift theorem,} \\
&= \int_\alpha f(\alpha)exp\{-i(2\pi u\alpha)\}d\alpha \int_x g(x)exp\{-i(2\pi ux)\}dx, \text{therefore,} \\
\mathcal{F}[f(x) * g(x)] &= F(u)G(u)
\end{aligned} \tag{7.10}$$

This result is commonly known as the convolution theorem and it states that convolution in the spatial domain is equivalent to multiplication in the frequency domain. It is easy to show that the opposite is also true; convolution in the frequency domain is equivalent to multiplication in

the spatial domain. As we will see, this result implies that convolution operators are essentially spectral filters whose bandpass characteristics are defined by their Fourier transforms.

7.4.3 Sampling Theorem - Aliasing

Figure 7.8 illustrates two spatial functions, $f(x)$ and $g(x)$. The first, $f(x)$ is a continuous spatial function representing the image function and the second, $g(x)$, is an infinite sequence of Dirac delta operators. The product of these two functions is a sampled approximation of the original $f(x)$.

$$h(x) = f(x)\sum_n \delta(x - nx_0) = \sum_n f(nx_0)\delta(x - nx_0)$$

We would like to determine the effects of the sampling function on the spectral energy in f(x). By the convolution theorem, we know that the product of these two spatial functions is equivalent to the convolution of their Fourier transform pairs.

$$\begin{aligned} f(x) &\xrightarrow{\mathcal{F}} F(u) \\ \sum_n \delta(x - nx_0) &\xrightarrow{\mathcal{F}} \frac{1}{x_0}\sum_n \delta(u - \frac{n}{x_0}) = G(u), \text{ and} \\ H(u) &= F(u) * G(u) \end{aligned}$$

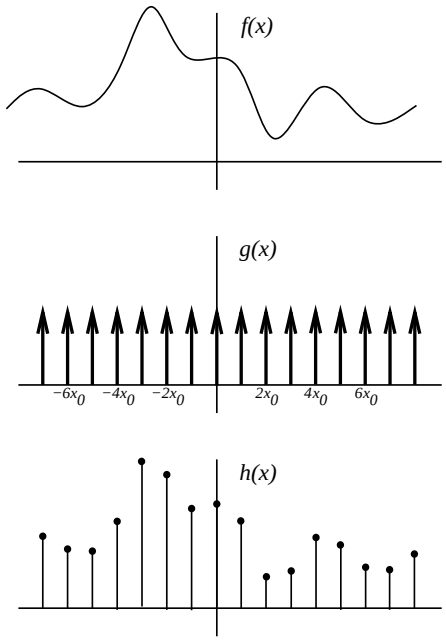

Figure 7.8 *A continuous spatial function and the sampling function.*

We may write the function $H(u)$ in terms of $F(u)$:

$$\begin{aligned} H(u) &= F(u) * \left[\frac{1}{x_0}\sum_n \delta(u - \frac{n}{x_0})\right] \\ &= \int_{-\infty}^{\infty} F(\alpha)G(u - \alpha) \\ &= \int_{-\infty}^{\infty} F(\alpha)\left[\frac{1}{x_0}\sum_n \delta(u - \alpha - \frac{n}{x_0})d\alpha\right] \\ &= \frac{1}{x_0}\int_{-\infty}^{\infty}\sum_n F(u - \frac{n}{x_0})\delta(u - \alpha - \frac{n}{x_0})d\alpha \\ &= \frac{1}{x_0}\sum_n F(u - \frac{n}{x_0})\int_{-\infty}^{\infty}\delta(u - \alpha - \frac{n}{x_0})d\alpha \\ H(u) &= \frac{1}{x_0}\sum_n F(u - \frac{n}{x_0}) \end{aligned}$$

Therefore, the frequency spectrum of the sampled image consists of duplicates of the spectrum of the original image distributed at $1/x_0$ frequency intervals. Figure 7.9 shows this effect schematically for two cases.

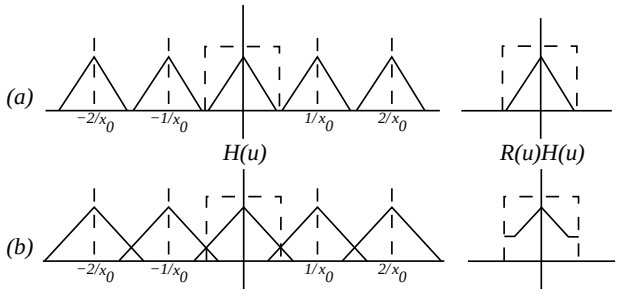

Figure 7.9 *The effects of sampling on reconstruction.*

$R(u)$ is a frequency domain bandpass filter

$$R(u) = \begin{array}{ll} 1 & if \quad |u| < 1/(2x_0), \\ 0 & otherwise \end{array}$$

Figure 7.9 clearly illustrates the conditions under which the function $f(x)$ can be reconstructed. When replicated spectra interfere, the crosstalk introduces energy at relatively high frequencies changing the appearance of the reconstructed image. The sampling theorem follows directly, *if the image contains no frequency components greater than one half the sampling frequency, then the continuous image is faithfully represented in the sampled image.*

7.4.4 The Discrete Fourier Transform and Convolution

A complex function $h(k1, k2)$ defined over a two-dimensional *spatial* grid $0 \leq k1 \leq (N_1 - 1),\ 0 \leq k2 \leq (N_2 - 1)$ can be described in the *spatial frequency* domain using the Fourier transform

$$H(n_1, n_2) = \sum_{k2=0}^{N_2-1} \sum_{k1=0}^{N_1-1} exp(2\pi i k_2 n_2/N_2) exp(2\pi i k_1 n_1/N_1) h(k_1, k_2) \tag{7.11}$$

where:

$$\begin{aligned} k_i &= \text{the sample spacing } [pixels] \\ n_i &= -\frac{N_i}{2}, \cdots, \frac{N}{2}, \text{ so that} \\ \frac{n_i}{N_i} &= \text{the discrete set of frequencies sampled} \end{aligned}$$

Note that the frequencies sampled in the transformation correspond exactly to the Nyquist Sampling rate;

$$-\frac{1}{2}\left[\frac{cycle}{pixel}\right] \leq f_n \leq \frac{1}{2}\left[\frac{cycle}{pixel}\right].$$

In Equation 7.11, the expression for $H(n_1, n_2)$ is periodic in both the spatial and frequency domain. This implies that, in one dimension, $H_{-n} = H_{N-n}$ or $H_{-N/2} = H_{N/2}$. This suggests that we can

let the n_i vary from 0 to $N_i - 1$ where the DC signal component (zero frequency) corresponds to $n = 0$, positive frequencies $0 < f < 1/2\frac{cycles}{pixel}$ correspond to $1 \leq n \leq N/2 - 1$, negative frequencies $-1/2\frac{cycles}{pixel} < f < 0$ correspond to $N/2 + 1 \leq n \leq N - 1$, and $n = N/2$ corresponds to both $f = 1/2\frac{cycles}{pixel}$ and $f = -1/2\frac{cycles}{pixel}$. Moreover, the complex exponential (Equation 7.11) can be decomposed dimensionally in Equation 7.11 suggesting that the 2D Fourier transform is equivalent to two 1D Fourier transforms applied sequentially, i.e. $\mathcal{F}_{k1}(\mathcal{F}_{k2}[h(k_1, k_2)])$.

The inverse discrete fourier is defined as:

$$h(k_1, k_2) = \frac{1}{N_1 N_2} \sum_{n2=0}^{N_2-1} \sum_{n1=0}^{N_1-1} H(n_1, n_2) exp(-2\pi i k_2 n_2 / N_2) exp(-2\pi i k_1 n_1 / N_1) \tag{7.12}$$

Consider the convolution of two functions, $g(x)$ and $f(x)$ (denoted $h = g * f$), where $g(x)$ is the one dimensional *image* function and $f(x)$ is the convolution operator or *response* function (Figure 7.10).

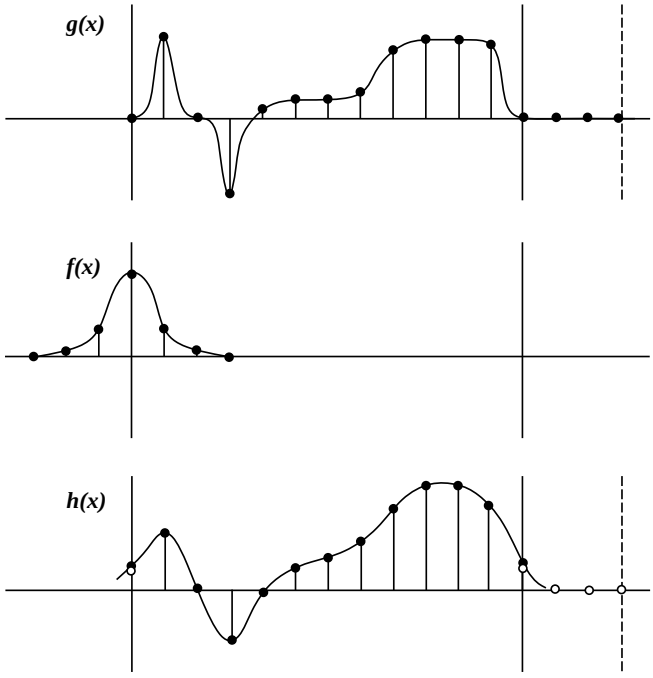

Figure 7.10 *The convolution of operator $f(x)$ with image function, $g(x)$.*

The effect of convolution is to map the function $g(x)$ to function $h(x)$ using a linear combination of intensities in a region of $g(x)$ defined by $f(x)$. If $f(x)$ was a unit impulse at $x = 0$, then $f(x)$ is just the identity filter leading to $h(x) = g(x)$. If $f(x)$ was the unit impulse at $x = 10$, then

$h(x) = g(x + 10)$. If, on the other hand, $g(x)$ consisted of the unit impulse or delta-function at $x = x_0$, the result $h(x)$ is a copy of $g(x)$ smeared into the shape of the convolution operator and shifted, i.e. $h(x) = f(x - x_0)$ for a symmetric operator.

Since both $g(x)$ and $f(x)$ are assumed to be periodic, when the operator is positioned near the beginning or end of any period in the signal, it can integrate signal data from the end or beginning of the signal, respectively. Figure 7.10 illustrates the effect; $h(x)$ is shown for the original 13 sample signal (filled circles) and for the *padded* 16 sample signal (open circles). The padding eliminates the *edge* effects introduced by the periodicity requirement.

EXAMPLE: Suppose that the rectangular function, $rect(x)$, is the convolution operator as in Figure 7.11. The Fourier transform of $rect(x)$ produces the $sinc(u)$ function (see Table 7.1). It therefore attenuates high frequencies while allowing the low frequencies to pass relatively unscathed. Sharp edges in the original image will become smooth and gradual in the filtered image. The spectral selectivity of $f(x)$ puts it in the class of so-called lowpass filters. Figure 7.11 shows the smoothing effect of $f(x)$ on a discrete test signal. As we would expect for a low pass filter, all rapid variation in the original image is removed in the convolution.

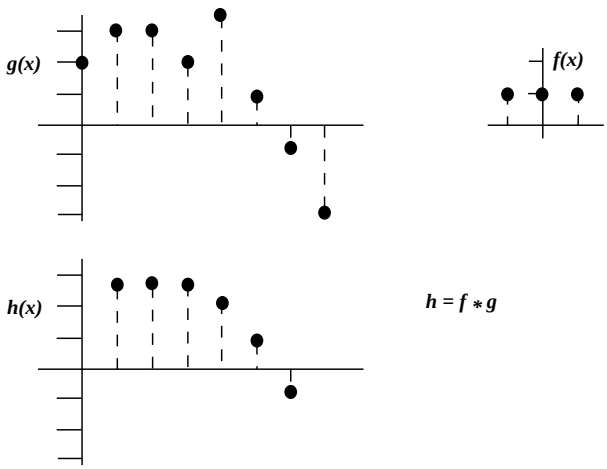

Figure 7.11 *A simple low pass filter by convolution.*

7.5 Early Processing

Many important clues about objects in the world can be determined by looking at small neighborhoods of pixels on the image plane. The human visual processing architecture devotes special purpose hardware to the task of identifying oriented edges, motion, texture, and many other features that depend only on local properties of the intensity function. This section will express this class of tasks mathematically in the form of a convolution with operators designed to identify important kinds of image features.

The convolution operation of a continuous function defined earlier using Equation 7.9 is repeated here for a two dimensional signal.

$$f(x,y) * g(x,y) = h(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u,v)g(x-u, y-v)dudv$$

For discrete functions, the equivalent operation is:

$$f(x,y) * g(x,y) = h(x,y) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} f(u,v)g(x-u, y-v)$$

or in a slightly more convenient form:

$$h(x,y) = \sum_{i=-\alpha}^{+\alpha} \sum_{j=-\alpha}^{+\alpha} f(i+\alpha, j+\alpha)g(x+i, y+j)$$

where, $h(x,y)$ is a new image generated by convolving the image $g(x,y)$ with the $(2\alpha + 1) \times (2\alpha+1)$ convolution mask, $f(i,j)$. Therefore, when α is 1, the convolution operator is a 3×3.

This is a convenient notation because it allows us to index the image relative to the center point of the convolution operator. The kind of processing represented by the convolution process is necessarily local, where a response $h(x,y)$ depends on a neighborhood of support in the original image $g(x,y)$ (Figure 7.12). Support is drawn from the image according to the definition of the convolution operator, $f(i,j)$.

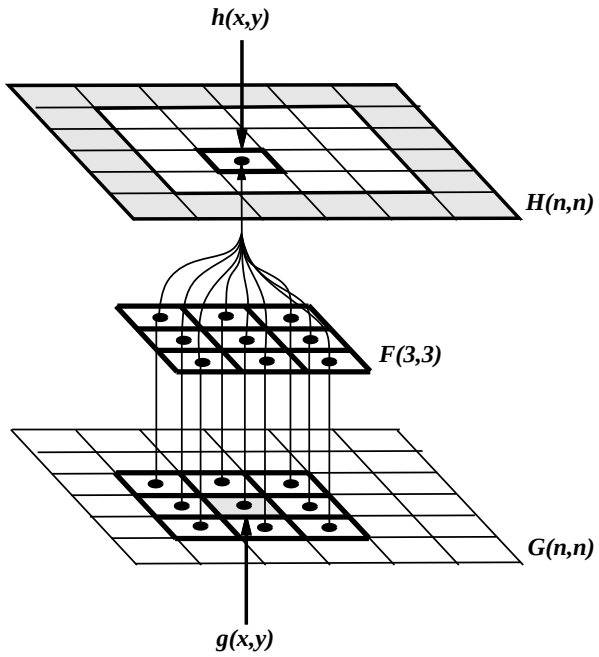

Figure 7.12 *Local Computation of Image Features.*

7.5.1 Edge Detection

> *"...there is evidence that the mammalian visual system responds to edges through special low-level template matching edge detectors."* — Hubel and Wiesel

Significant gradients in the intensity function are highly correlated with spatial discontinuities in the world. This is obvious when one considers the intensity variation between an object and its background along the occluding contour, but it is also the case when a continuous surface quickly changes the direction of its normal (a region of high local curvature). For this reason, locations on the image plane with steep intensity gradients are often referred to as *edges*.

Intensity Gradients

The gradient for a two dimensional image function, $g(x,y)$, can be computed directly if one can estimate the value of dg/dx and dg/dy.

$$\nabla g(x,y) = \frac{dg}{dx}\hat{x} + \frac{dg}{dy}\hat{y} \tag{7.13}$$

One way of estimating the derivatives is by using a finite difference approximation:

$$\frac{dg(x,y)}{dx} \approx \frac{g(x+1,y) - g(x-1,y)}{2} \qquad \frac{dg(x,y)}{dy} \approx \frac{g(x,y+1) - g(x,y-1)}{2}$$

As a finite difference, the gradient clearly depends only on local information on the image plane, and can be conveniently expressed as a convolution with an appropriate convolution operator. For the derivative in the x direction, $f_x = [-1/2,\ 0,\ 1/2]_{1\times 3}$, and in the y direction, $f_y = [-1/2,\ 0,\ 1/2]^T_{3\times 1}$. The magnitude of the intensity gradient and its orientation on the image plane can likewise be computed.

$$|\nabla g(x,y)| = \left[\left(\frac{dg}{dx}\right)^2 + \left(\frac{dg}{dy}\right)^2\right]^{\frac{1}{2}} \tag{7.14}$$

$$\phi(x,y) = tan^{-1}\left(\frac{dg/dy}{dg/dx}\right) \tag{7.15}$$

A number of edge operators often used in the literature are based on this simple finite difference approximation (see Table 7.2). However differentiation, in general, tends to amplify the effects of noise in the signal and so is typically preceded by a lowpass filtering operation. While this may at first seem contradictory, if we assume that the useful information is a bandlimited signal (ultimately, it must be due to the sampling density on the image plane), and further that the noise is predominately high frequency, then the preliminary low pass filter will smooth the image with marginal degradation in the information content, while dramatically reducing noise in the gradient computation.

Table 7.2 presents some commonly used gradient operators that accomplish smoothing and differentiation simultaneously - smoothing by virtue of averaging the gradient computation over several rows or columns, and differentiation by the finite difference operator.

Edge detection can be based directly on the magnitude and orientation of the gradient vector. However, if we define an edge by specifying that the gradient magnitude must exceed a threshold, then small thresholds find many faint edges (which may be good) while "blooming" strong edges over several pixels. High thresholds mean sharper edges but far fewer of them.

Table 7.2: Gradient (first derivative) Operators

operator	∇_1	∇_2
Roberts	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Prewit	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Edge Sharpening

It would seem valuable, in general, to detect relatively faint edges while at the same time maintaining precision at strong edges. One way of accomplishing this is to require that the second derivative of the intensity function be near zero when the first derivative is above threshold. In Figure 7.13, if a simple threshold on the gradient magnitude where employed, then the edge would appear to be five pixels wide. If, however, we look for those pixels near zero in the second derivative which also have gradient magnitudes greater than the threshold, then even for these strong intensity gradients can lead to precise edge locations. This is equivalent to asserting that the inflection point in the intensity function is a good estimate of the *center* of the edge.

The **Laplacian** operator approximates the second derivative of the image function:

$$\nabla^2 g = \frac{d^2 g}{dx^2} + \frac{d^2 g}{dy^2} \Rightarrow f = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

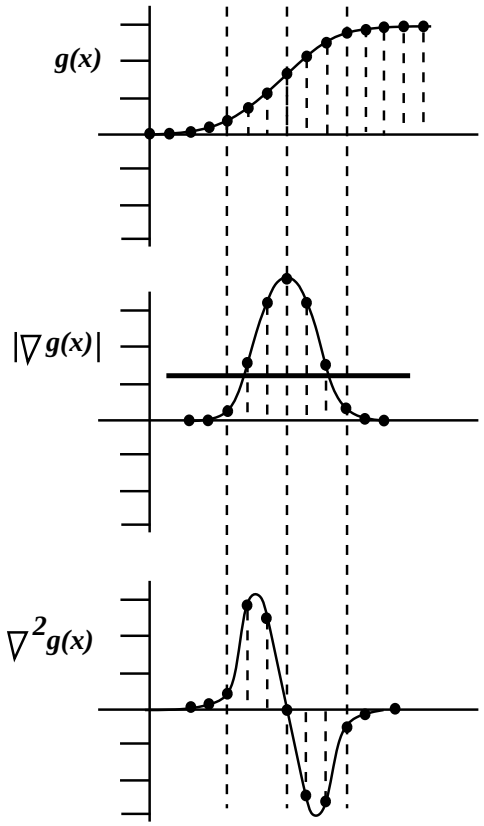

Figure 7.13 *Identifying the inflection point in the intensity function.*

The Laplacian operator is often used to identify locations where there is a sign change in the Laplacian image. The zero crossings of the the resulting image can be used to estimate the apparent center of an edge whose gradient is significant over several pixels.

If we look for adjacent pixels where the sign of the second derivative changes, then we can interpolate between them and locate zero crossings to floating point precision (sometimes called sub-pixel accuracy).

7.5.2 Gaussian Operators

Gaussian convolution operators are derived from the Gaussian function

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{7.16}$$

The Gaussian operator is really a family of operators parameterized by σ which is the *scale* of the Gaussian. Increasing the scale by increasing σ dilates the function, so that the filter response draws support from a larger region of the image plane, Figure 7.14(A) shows the Gaussian operator at three scales. This operator behaves as a lowpass filter with a cutoff frequency that is inversely proportional to σ.

It can be shown (homework problem 2a at the end of this chapter) that for any two functions, f and g, $f * g' = (f * g)'$. This implies that using g_σ to smooth the image and then differentiating, is equivalent to convolving with g'_σ.

$$g'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-x^2/2\sigma^2} \tag{7.17}$$

The same can be said for the second derivative:

$$g''_\sigma(x) = \frac{1}{\sqrt{2\pi}} \left[\frac{x^2}{\sigma^5} - \frac{1}{\sigma^3} \right] e^{-x^2/2\sigma^2}. \tag{7.18}$$

Figure 7.14(B) and (C) show the first and second derivatives of the Gaussian operator at the same three scales. These functions are similar to the finite difference operators we saw earlier. The first derivative of the Gaussian is symmetric and odd. It also behaves like an edge detector with bandpass characteristics that are dependent on σ. In one dimension, the second derivative of the Gaussian is equibvalent to the Laplacian operator and is likewise parameterized by σ. This family of operators (Gaussian and its derivatives) can be used, therefore, to detect and localize edges in much the same way as say the Sobel operator and the Laplacian. Moreover, having been derived from the Gaussian, they combine smoothing with differentiation.

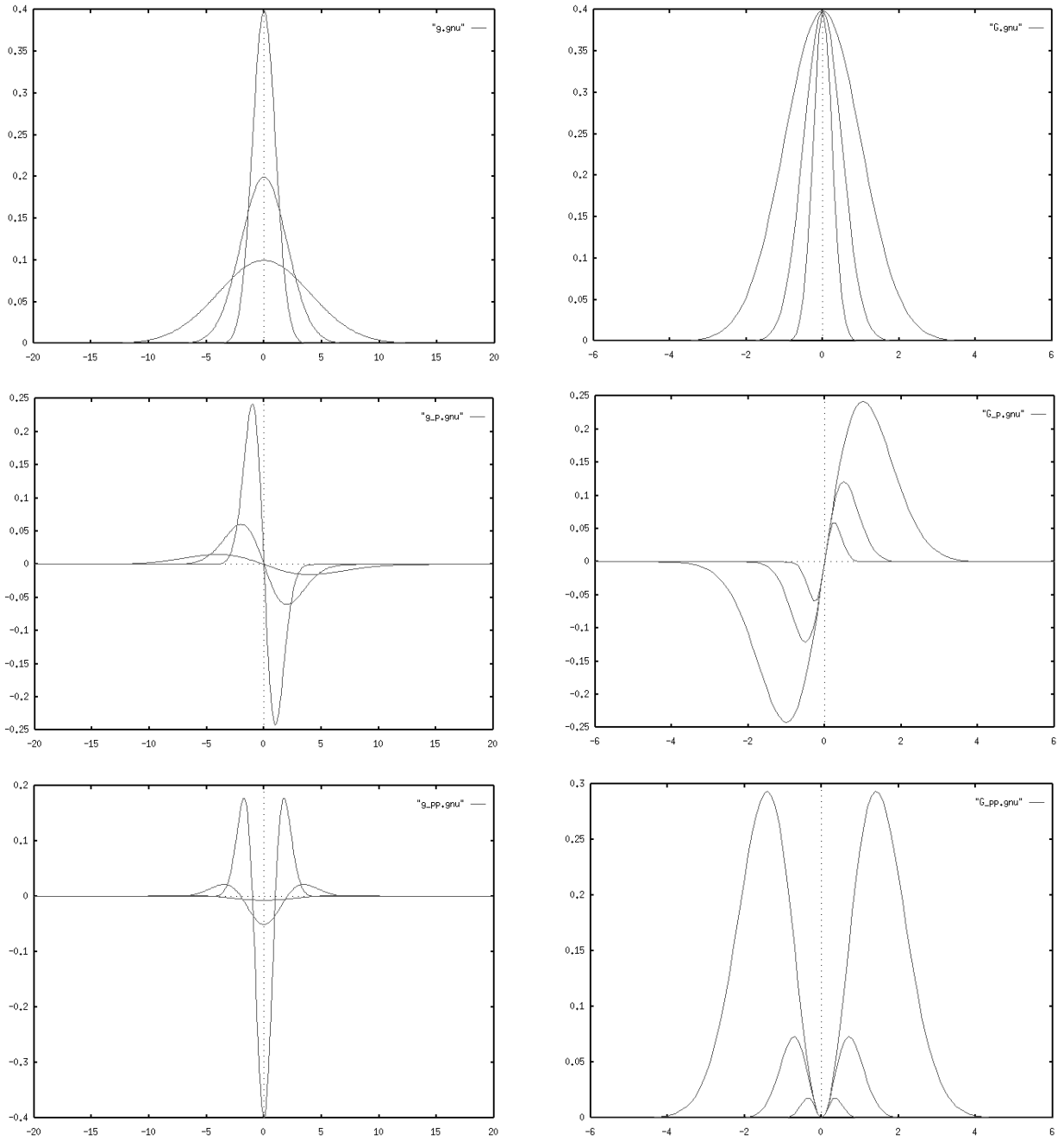

Figure 7.14 *Fourier transform pairs for derivative of Gaussian operators: (A) $g(x)$ and $G(w)$ for $\sigma = 1, 2, 4$, (B) $g'(x)$ and $G'(w)$ for $\sigma = 1, 2, 4$, (C) $g''(x)$ and $G''(w)$ for $\sigma = 1, 2, 4$.*

To detect vertical edges, for instance, one may choose to convolve with $G'_\sigma(x)G_\sigma(y)$. The effect is to smooth in both the x and y directions, and to differentiate in the x direction.

Center Frequency

From Table 7.1 for the Gaussian:

$$Ae^{-px^2} \xrightarrow{\mathcal{F}} \frac{A}{\sqrt{2\pi}} e^{w^2/4p}$$

If we apply this relation to Equation 7.16, we find

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \xrightarrow{\mathcal{F}} \frac{1}{\sqrt{2\pi}} e^{-w^2\sigma^2/2} = G(\omega).$$

Further, by making use of another property of the Fourier transform (Table 7.1), namely

$$\mathcal{F}\left[\frac{d^n}{dx^n} f(x)\right] = (i\omega)^n F(\omega),$$

and ignoring phase information,

$$g'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-x^2/2\sigma^2} \xrightarrow{\mathcal{F}} \frac{\omega}{\sqrt{2\pi}} e^{-w^2\sigma^2/2} = G'(\omega)$$

and,

$$g''_\sigma(x) = \frac{1}{\sqrt{2\pi}} \left[\frac{x^2}{\sigma^5} - \frac{1}{\sigma^3}\right] e^{-x^2/2\sigma^2} \xrightarrow{\mathcal{F}} \frac{\omega^2}{\sqrt{2\pi}} e^{-w^2\sigma^2/2} = G''(\omega)$$

The right column of Figure 7.14 plots these frequency domain representations of the Gaussian operators. To solve for the center frequency for the derivative of a Gaussian, we find the extremum in $G^n(\omega)$.

$$\frac{\partial}{\partial\omega} \left[(i\omega)^n \; G(\omega)\right]_{\omega=\omega_0} = 0$$

ω_0	$n=0$	1	2
$\sigma = 1$	0	1	$\sqrt{2}$
2	0	1/2	$\sqrt{2}/2$
4	0	1/4	$\sqrt{2}/4$

The maximum in the function is found by looking for a slope of zero in the $G^n(\omega)$ function:

$$\begin{aligned} (i^n)(n\omega_0^{n-1} G(\omega_0) + (i\omega)^n \frac{\partial}{\partial\omega_0} G(\omega_0) &= 0 \\ (i^n)(n\omega_0^{n-1} + (i\omega_0)^n(-\sigma^2\omega_0) &= 0 \\ n\omega_0^{n-1} &= \sigma^2\omega_0^{n+1}, \end{aligned} \tag{7.19}$$

$$\text{so that, } \omega_0 = \frac{\sqrt{n}}{\sigma} \tag{7.20}$$

The table above summarizes the center frequency results for the functions in the right column of Figure 7.14.

Equivalent Rectangular Bandwidth

Many methods exist with which to approximate the bandbass characteristics of the Gaussian operators. The equivalent rectangular bandwidth equates the area under the energy density function $|G_n(\omega)|^2$ to the rectangular bandpass filter with height $|G_n(\omega_0)|^2$ and width $2W$.

$$(2W)(|G_n(\omega)|^2) = \int_0^\infty |G_n(\omega)|^2 d\omega$$

Therefore,

$$\begin{aligned} W &= \frac{\int |G_n(\omega)|^2 d\omega}{2|G_n(\omega_0)|^2} \\ &= \frac{\int |(i\omega)^n e^{-\omega^2\sigma^2/2}|^2 d\omega}{2|(i\omega)^n e^{-\omega^2\sigma^2/2}|^2_{\omega=\omega_0=\sqrt{n}/\sigma}} \\ &= \frac{\int (\omega)^{2n} e^{-\omega^2\sigma^2} d\omega}{2(\sqrt{n}/\sigma)^{2n} e^{-n}}. \end{aligned}$$

To evaluate the integral in the numerator, we introduce a change of variable:

$$\begin{aligned} & z = \omega^2\sigma^2 \\ & dz = \sigma^2(2\omega)d\omega, \qquad \text{so that} \\ & \int (\omega)^{2n} e^{-\omega^2\sigma^2} d\omega = \frac{1}{2\sigma^{2n+1}} \int z^{n-1/2} e^{-z} dz \end{aligned}$$

and we may write the rectangular bandwidth for the Gaussian in terms of the gamma function,

$$W = \frac{e^n}{4\sigma n^n}\Gamma(n + \frac{1}{2}), \tag{7.21}$$

where,

$$\Gamma(n+1) = \int_0^\infty x^n e^{-x} dx$$

Equation 7.21 defines the equivalent rectangular bandwidth for n^{th} order derivatives of the Gaussian operator with scale σ. To evaluate it numerically, we will make use of some important properties of the gamma function, namely:

$$\begin{aligned} \Gamma(n+1) &= n\Gamma(n) \\ \Gamma(\frac{1}{2}) &= \sqrt{2\pi}. \end{aligned}$$

The value of W can then be found using the following expressions:

ω_0	$n=0$	1	2
$\sigma=1$	0.6267	0.8517	0.8682
2	0.3133	0.4259	0.4342
4	0.1567	0.2129	0.2170

$$n=0 \qquad W = \frac{1}{4\sigma}\Gamma(\frac{1}{2}) = \frac{\sqrt{2\pi}}{4\sigma}$$

$$n=1 \qquad W = \frac{e}{4\sigma}\Gamma(\frac{3}{2}) = \left(\frac{e}{4\sigma}\right)\frac{1}{2}\Gamma(\frac{1}{2}) = \frac{e\sqrt{2\pi}}{8\sigma}$$

$$n=2 \qquad W = \frac{e^2}{16\sigma}\Gamma(\frac{5}{2}) = \left(\frac{e^2}{16\sigma}\right)\frac{3}{2}\Gamma(\frac{3}{2})$$
$$= \left(\frac{3e^2}{32\sigma}\right)\frac{1}{2}\Gamma(\frac{1}{2}) = \frac{3\sqrt{2\pi}e^2}{64\sigma}$$

These bandwidth results are tabulated for the spectra illustrated in the right hand column of Figure 7.14.

7.5.3 Motion

7.5.4 Texture

7.5.5 Template Matching and Normalized Cross-Correlation

Many times there are specific features that we are interested in locating on the image plane. Consider the visual control of an automobile. In such an application, it seems an effective policy to identify lane markers, traffic signs, and perhaps other vehicles. One could go about establishing a complex conjunction of features that indicate a stop sign, for instance, but if we already know precisely what a stop sign looks like, then we can construct a more specialized visual template and look for instances of it in the image. This is a generalization of the techniques for finding lower level features that we discussed earlier. For example, the edge operators listed in Table 7.2 can be viewed as a prototypical edges — relatively high intensities on one side and relatively low intensities on the other. If we wish, we may think of these templates for edges rather than as finite difference operators. From this perspective, the convolution of an image with these edge templates constitutes a pixel-wise product of the template pattern with the image.

This process is just the convolution

$$h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} t(\alpha,\beta)g(x+\alpha, y+\beta)d\alpha d\beta$$

where $t(\alpha,\beta)$ is the template (indexed relative to the center pixel) and $g(x+\alpha, y+\beta)$ is the relevant region of the image around location (x,y). We may write it as the discrete cross correlation of g

and t

$$R_{gt} = \sum_{\alpha} \sum_{\beta} t(\alpha, \beta) g(x + \alpha, y + \beta).$$

The cross correlation is maximized when the intensity function g in the region around location (x, y) is *shaped-like* the template function t. Maxima in R are minima in the sum of squared differences between the template and the image region

$$\sum_{i=-\alpha}^{\alpha} \sum_{j=-\beta}^{\beta} [g(x + i, y + j) - t(\alpha, \beta)]^2$$

This is another example of an early processing technique that may be used to highlight image features that we are interested in such as edges, or textures, and more general templates like corners of a particular orientation, or the stop sign discussed earlier. However, the result can be sensitive to variations in ambient brightness, occlusion, shadows, scaling and perspective distortion.

The vector normalized difference operator is somewhat insensitive to contrast, brightness, lighting and other uncontrollable characteristics of the image formation process. In this technique, both the template t and the region aroundf $g(x, y)$ are *normalized* by first subtracting the mean intensity and then normalizing the magnitude of the resulting function.

The correlation between a normalized template and a normalized image is given by:

$$R(x, y) = \sum_{i=-\alpha}^{\alpha} \sum_{j=-\beta}^{\beta} \left[\left(f(x + i, y + j) - \hat{f} \right) \left(t(i + \alpha, j + \beta) - \hat{t} \right) \right] / (VW)$$

where, $-1 \leq R(x, y) \leq +1$, is the normalized correlation of the $(2\alpha + 1) \times (2\beta + 1)$ template to the image at image location (x, y). This correlation depends on (constant) properties of the template:

$$\hat{t} = \left[\sum_{i=-\alpha}^{\alpha} \sum_{j=-\beta}^{\beta} t(i + \alpha, j + \beta) \right] / (MN)$$

where $M = (2\alpha + 1)$ and $N = (2\alpha + 1)$ represent the dimensions of the template, and

$$W = \left[\sum_{i=-\alpha}^{\alpha} \sum_{j=-\beta}^{\beta} (t(i + \alpha, j + \beta) - \hat{t})^2 \right]^{1/2}.$$

The correlation metric also depends on the properties of the image in the region about location (x, y):

$$\hat{f} = \left[\sum_{i=-\alpha}^{\alpha} \sum_{j=-\beta}^{\beta} f(x + i, y + j) \right] / (MN)$$

where $M = (2\alpha + 1)$ and $N = (2\alpha + 1)$ represent the dimensions of the template, and

$$V = \left[\sum_{i=-\alpha}^{\alpha} \sum_{j=-\beta}^{\beta} (f(x+i, y+j) - \hat{f})^2 \right]^{1/2}.$$

7.6 Segmentation

7.6.1 Edge Relaxation

The relaxation method is based on heuristics capturing the probability of seeing edges of various configurations on the image plane. In a neighborhood defined by three pixels on either side of the edge, four types of terminators are identified for the central edgelet: type 0, 1, 2, and 3. Consider the two neighborhoods of the image plane shown in the Figure 7.15. Forward difference gradients in the easterly and southerly directions are computed for every pixel in the original image plane. For south gradients (gradients in the $\hat{i}$ direction), then the left-right neighborhood is defined about the south gradient in question. Top-bottom neighborhoods are constructed about east gradients in a similar fashion.

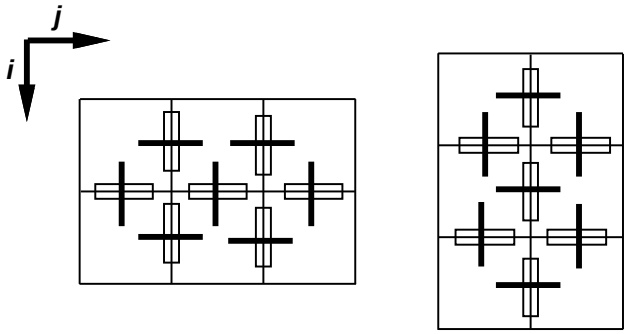

Figure 7.15 *Left-right and top-bottom neighborhoods for a south and east edge gradient, respectively.*

The edge relaxation procedure involves iterating over the east and south gradient images looking for evidence supporting the central edge hypothesis. That is, if the central gradient is in the southerly direction, then we look at the left and right neighborhoods for edges that could be the continuation of a line containing the central edge. Type 0 neighborhoods are simply terminators with no evidence of significant intensity gradients adjacent to the central edge. Type 1, 2, and 3 neighborhoods correspond to 1, 2, and 3 significant adjacent edgelets, resectively. The 4 types of terminators are illustrated in the following Figure 7.16.

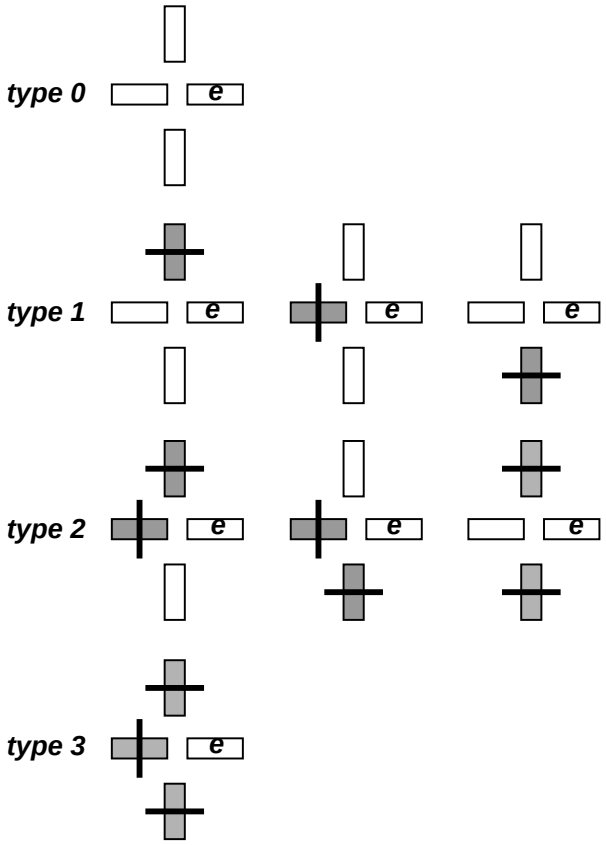

Figure 7.16 *The unique types of a left-neighborhood.*

Classifying edge types is accomplished by comparing the gradients at neighborhood pixels on a *relative* scale to identify the dominant neighborhood characteristics. If we define q as the absolute cutoff for significant gradients, then the dominant neighborhood type can be estimated as follows:

$$\begin{aligned} conf[0] &= (m - neighbor[a])(m - neighbor[b])(m - neighbors[c]) \\ conf[1] &= neighbor[a](m - neighbor[b])(m - neighbor[c]) \\ conf[2] &= neighbor[a]neighbor[b](m - neighbor[c]) \\ conf[3] &= neighbor[a]neighbor[b]neighbor[c] \end{aligned}$$

where: (a, b, c) is a sorted list of the gradient magnitudes, $a > b > c$, $q = constant$, and $m = max(a, b, c, q) = max(a, q)$. The maximum confidence value computed identifies the dominant neighborhood type.

Conjunctions of neighbor types over left-right or top-bottom neighbors can be used to express

heuristics that affect the confidence of the central edge. These heuristics are captured in the following table.

$$\text{confidence[left-type][right-type]} = \begin{array}{cccc} -\delta & 0 & -\delta & -\delta \\ 0 & \delta & \delta & \delta \\ -\delta & \delta & 0 & 0 \\ -\delta & \delta & 0 & 0 \end{array}$$

7.6.2 Hough Transform

generalized hough transform

7.7 Binocular Imaging

The process of projecting the irradiance function onto the image plane disgards a great deal of information. It is not, in general, possible to reconstruct the 3D structure of the world with a single image — at least two images, acquired simultaneously or in a temporal sequence, are required to locate a feature in 3D. The geometry of a simple binocular vision system is presented in Figure 7.17. The geometry of the imaging system can be used to solve for the world frame coordinate of feature point P.

Referring to the diagram of the $x-z$ plane in Figure 7.17, we define $\phi_L = atan(\frac{u_L}{f})$, $\phi_R = atan(\frac{u_R}{f})$, define $\gamma_L = \theta_L + \phi_L$ and $\gamma_R = \theta_R + \phi_R$, and the perspective vector (from the camera focal point to the feature) as $(cos(\gamma_L), sin(\gamma_L))$ and $(cos(\gamma_R), sin(\gamma_R))$, respectively for the left and right cameras. We may write the kinematic loop equations for x and y:

$$\begin{aligned} \lambda_L cos(\gamma_L) - d &= \lambda_R cos(\gamma_R) + d \\ \lambda_L sin(\gamma_L) &= \lambda_R sin(\gamma_R), \end{aligned}$$

where $\lambda_{L,R}$ are the lengths of the feature vectors oriented along the normalized perspective vectors for the left and right camera, respectively.

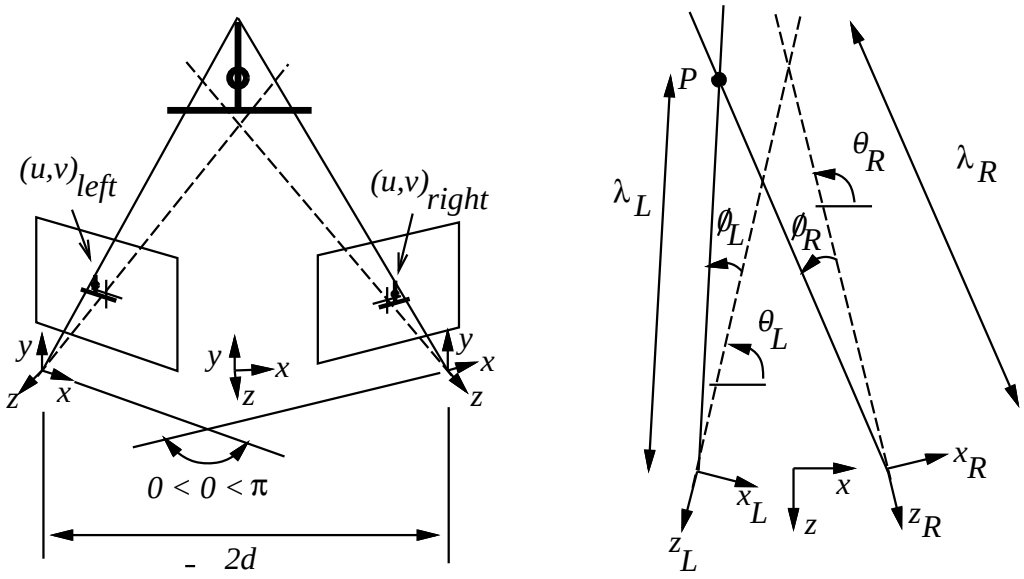

Figure 7.17 *The binocular imaging geometry*

We have two equations in two unknown parameters, $\lambda_{L,R}$. It is a simple exercise to solve these relations to yield:

$$\begin{aligned} \lambda_L &= \frac{2dsin(\gamma_R)}{sin(\gamma_R - \gamma_L)} \\ \lambda_R &= \frac{2dsin(\gamma_L)}{sin(\gamma_R - \gamma_L)} \end{aligned}$$

These parameters together with kinematic equations can now be used to solve for the feature coordinates in the world frame. Notice that the relative *vergence* of the two feature vectors, $\gamma_R - \gamma_L$ is derived from two different sensory modalities. The kinematic configuration of the camera measures the mechanical convergence of the binocular system, and the offset from image center to feature measures the angular error from the camera perspective vector to the feature vector for both cameras. Combining the binocular configuration information with the image plane coordinate of the feature, there is enough information to reconstruct the 3D geometry of point(s) in the world.

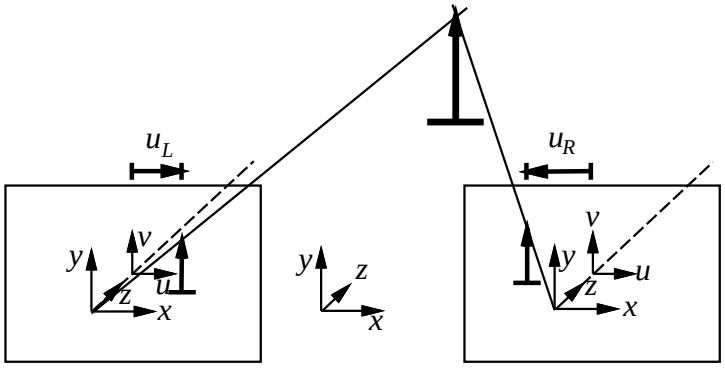

Figure 7.18 *Disparity encoded depth in a zero vergence binocular configuration*

EXAMPLE: Consider the simple binocular configuration illustrated in Figure 7.18. In this geometry, the stereo system encodes depth entirely in terms of disparity. Under these conditions

$$u_L = \frac{f(x-d)}{z} \qquad u_R = \frac{f(x+d)}{z}$$

$$\begin{aligned} zu_R &= f(x+d) \\ zu_L &= f(x-d) \end{aligned}$$

$$z(u_R - u_L) = 2df$$

So by eliminating x , we may solve directly for z

$$z = \frac{2df}{(u_R - u_L)}$$

However, 3D reconstruction will prove to be much more challenging than this simple geometric construction suggests. The real problem is identifiying features in the right image plane that correspond to features in the left image plane. In other words, how can we make sure that the feature we look at with the right eye is the same as the feature we are looking at with the left eye?

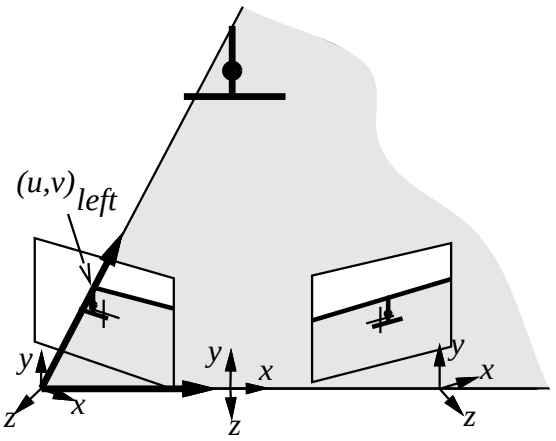

Figure 7.19 *The epipolar constraint*

We might formulate the solution to the correspondance problem as a search over the left image plane for an intensity function that is highly correlated to a region of interest on the right image plane. This would be extremely expensive without more to go on. Fortunately, geometry can simplify this search somewhat. Figure 7.19 illustrates how when given a feature in the left image, one may constrain the search for its counterpart in the right image plane. The plane formed by the line joining the focal points of the stereo system and the ray from the left focal point outward, through the image feature of interest contains the 3D feature and its projection onto both left and right image planes. Therefore, when searching for a region on the right image plane that is highly correlated to a given region of the left image plane, one must only search along the line of intersection between the so-called *epipolar* plane and the right image plane.

7.8 Visual Servoing

References

Hanson, Weiss, Oliensis, Kohl,T. and Kohl, C. , *VisionTutor Lecture Guide, Version Beta 1.0, August 5, 1992.*

Oliensis, Thomas *CS-TR, University of Massachusetts.*

Pratt , *Digital Image Processing, John Wiley and Sons, New York, 1978.*

Feynman, Leighton, and Sands , *The Feynman Lectures on Physics, Addison-Wesley, 1966.*

Horn , *Robot Vision, MIT Press, Cambridge MA, 1986.*

Young, T.Y. and Fu, K.S. , *Handbook of Pattern Recognition and Image Processing, Academic Press, New York, 1986.*

Ballard and Brown , *Computer Vision, Prentice Hall, Englewood Cliffs, NJ, 1982.*

Marr , *Vision, W.H. Freeman, San Francisco, 1982.*

Nevatia , *Machine Perception, Prentice-Hall, Englewood Cliffs, NJ, 1982.*

Rosenfeld and Kak , *Digital Picture Precessing - Vols. 1 and 2, Academic Press, New York, 1982.*

Hall, E. , *Computer Image Processing and Recognition, Academic Press, New York, 1979.*

Hanson and Riseman , *Computer Vision Systems, Academic Press, New York, 1978.*

Levine , *Vision in Man and Machine, McGraw-Hill, New York, 1985.*

7.9 Homework Exercises

1. **Histogram Equalization**
   Design and implement an algorithm to equalize the image histogram. Plot the histogram before and after equalization and submit the corresponding images. See if you can find images on which equalization works poorly.

2. **Convolution**

   (a) Show that for any two functions, f and g, $f * g' = (f * g)'$.

   (b) Write a procedure, $convolve(f, g, h)$ that convolves image g with convolution mask f yielding the filtered image h. Use the Soble operator on test images. Show what the gradient magnitude image looks like on several images and explain the results.

   (c) The Laplacian operator approximates the second derivative of the image function.

      i. Derive the Laplacian operator as it appears in the notes using finite differences.

      ii. Convolve a selected image with the Laplacian operator and identify pixels where there is an east or south sign change in the Laplacian image. Show these zero crossings of the the resultng image.

      iii. Use your favorite gradient operator and prune out weak edges. Show the results for a variety of gradient thresholds.

3. **Edge Relaxation**
   Implementing an iterative relaxation procedure to enhance the interpretation of edgelets in and image by propagating corroborating information across the image plane. This involves scanning the image and at each pixel compute the east and south gradient to adjacent pixels. For each easterly gradient, evaluate the type of its top-bottom neighborhoods and for each southerly gradient, evaluate the left-right neighborhood types. After determining the neighborhood types, increment or decrement the edge *confidence* using the heuristics in the text. Continue this process until the confidence stabilizes everywhere across the image. The effect should be to fill in spurious gaps in the edges while eroding edges arising principally from noise. Test your procedure on several images in the image library and report your results.

4. **Fourier Transform**
   Demonstrate the convolution theorem using the Fast Fourier Transform (FFT) supplied as part of the LLVS vision environment. A stub for the solution is provided in *fft_convolve.c*. Pick a convolution operator and show that the convolution $(g * f)$ produces the same result as $\mathcal{F}^{-1}[\mathcal{F}[g] \times \mathcal{F}[f]]$. How does the placement of operator f within the image array effect the result? Discuss wrap-around or edge effects, how can you avoid them?

5. **Spatial Frequency Filter**
   Plot the *power* spectrum $(H(n_1, n_2)^2)$ for sample images. Implement a bandpass filter by setting $H(n_1, n_2) = 0$ for spatial frequencies greater that n_{cutoff}/N (low pass), and less than n_{cutoff}/N (high pass). Show the resulting image and its power spectrum.

6. **Template Matching**
   Demonstrate the performance of normalized cross correlation for matching *templates* in the image plane. Use the *xv* program to grab images from anywhere you wish that exhibit repeated occurances of a local image feature. An example of such an image and template is the *circles.pgm* image and the $circle_template.pgm$ image in the $cs603/xv_vision/images/$ directory. Locate the position of the template in the image by normalized cross correlation and write your template into the image at those image coordinates that appear to be good matches, i.e. those locations in the image where $|R(x, y)| \geq threshold$.

7. **Hough Transform**
   Use the generalized Hough transform to accumulate evidence for the existence of circles in image *circles.pgm*.

   (a) Design a table for casting votes in the Hough accumulator based on the edges (magnitude, orientation, or both) in the prototypical circle in *circle_model.pgm*.

   (b) Find edges in *circles.pgm* and use the generalized Hough transform to generate a Hough accumulator array for the prototype circle. Show the result as an image.

(c) Threshold the resulting Hough accumulator array to identify locations where the circle hypotheses have significant support. Indicate these possible locations for the circle prototype in the original image by superimposing a small marker at the likely center of the circle.

8. **Edge Relaxation**
Use the edge relaxation technique described in the text to link edges into contiguous lines. Try your code on a natural image — be advised that you should probably debug your code on small segments of the image which should produce predictable results. Show how local evidence propagates throughout the image to yield physically meaningful interpretations. In your write up, show some of the intermediate edge interpretations which arise while the procedure converges.

9. **Stereo Reconstruction**
Reconstruct a spatial representation $range(x, y)$, containing the z-coordinate of objects in the stereo pair *castle_left.pgm* and *castle_right.pgm*. This image pair can be found in the *xv_vision/images* directory in the *cs603* common directory. Choose the cycloptic coordinate frame for this representation so the disparity equation for depth discussed in class works (you can compare your results to ground truth if you wish, but this takes a great deal of additional work, so it is optional, but some presentation of your results is required). All spatial dimensions are given in *mm*. In the images directory, you will find:

$castle*.pgm$	–	pgm formatted images
$castle*.par$	–	image parameters
castlepoints.txt	–	text description of the ground truth points
castlepoints.xyz	–	world coordinates of the ground truth
$castle*.gt$	–	image coordinates where ground truth is reported

There is a README file in this directory with more detail. The image geometry is arranged so that corresponding rows in the images can be used roughly as epipolar lines.

Construct the depth map by extracting a neighborhood in the left image and searching the corresponding row in the right image for the maximum normalized cross correlation. Compute the disparity for this correspondance and use the disparity relation discussed in class to solve for the spatial coordinates of this point in space. Do this over an entire sub-image that you designate. The z coordinate of the result can be saved in an array indexed by the x and y coordinate of the left image plane. Remap the result into ranges between $0 - 255$ and write the result as a pgm image. Illustrate and comment on your results.