1 Recovery of Image Descriptions

To make image analysis practical, we must reduce the huge size of most image data. *Edge detection* and *segmentation* are important data reduction steps in most recovery algorithms. An *edge* in an image is an image contour across which the brightness of the image changes abruptly. Edges are classified by the way the brightness function [1] is distributed around the edge: *step edges*, *roof edges* and *line edges*. A *step edge* refers to a contour in the image across which the value of the intensity function changes abruptly. A *roof edge* is a contour across which the orientation of the intensity function changes abruptly. A *line edge* is a pair of adjacent parallel step edges. Our focus here will be on step edges.

1.1 Edge Detection

Edges in an image are important because they encode information about the structure of the scene. In a typical image, we can distinguish several physical events in a scene that lead to intensity edges in an image of the scene. Among these events are discontinuities in the surface normal, in illumination, in depth and in surface reflectance or in some combination of these.

Edge detection is a two-step process: first, short linear edge segments are detected, and then these edges are aggregated into extended segments. Algorithms for detecting edges (edge detectors) are usually differentiation-based or model-based. Differentiation approaches estimate the derivatives of the image intensity function, the idea being that large image derivatives reflect abrupt intensity changes. Model-based approaches try to determine whether the intensities in a small area conform to some model for the edges that we have assumed.

1.2 Differentiation Approaches

The gradient of a function $g(x, y)$ is the vector $\bigtriangledown g(x, y) = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}\right)$. At a step edge, the underlying intensity function has a large gradient pointing across the edge. This is the key idea behind most differentiation approaches to edge detection.

In most techniques, an image point is selected as representing an edge if the image intensity gradient at that point is above a certain threshold. The difference operators described in this section are discrete approximations of the gradient. There are many ways to estimate gradients. Most of the known approaches estimate the directional derivatives of the image intensity at any two orthogonal directions in a single point. If g_1 and g_2 are these orthogonal directional derivatives, then the magnitude of the gradient is $\sqrt{g_1^2 + g_2^2}$ and its direction with regard to g_1's direction is $\tan^{-1}(g_2/g_1)$.

If we consider a 2×2 window in the image and fit a planar surface to the values of the image intensity, then a good approximation to the gradient in the region defined by the

[1] Recall that what is available to a computer for the purposes of image understanding is a digital image. This digital image, however, has meaning in the optical image that it represents. The optical image is characterized by the image irradiance function defined over the image plane, and thus we can talk about the characteristics of the continuously defined intensity surface underlying a digital image.

(a)

g_1		g_2	
0	1	1	0
−1	0	0	−1

(b)

g_1		g_2	
−1	1	1	1
−1	1	−1	−1

Figure 1: **Roberts operators**: (a) the indicated masks are used to compute g_1 and g_2 at a 2×2 window, along the lines diagonal to the coordinate axes of the image, (b) the indicated masks are used to compute g_1 and g_2 at a 2×2 window along the x and y coordinate axes of the image.

g_1

−1	0	1
−1	0	1
−1	0	1

g_2

1	1	1
0	0	0
−1	−1	−1

Figure 2: **Prewitt operator**: these masks provide estimates of the directional derivatives along the coordinate axes of the image at the center of a 3×3 window.

window is the gradient of the fitted plane. A discrete approximation to the gradient based on this ideas can be computed using *Roberts operators* as shown in Figure 1.

We describe operators in terms of either 2×2 or 3×3 arrays of numbers called *masks* that are used to compute approximations to the directional derivatives at a point. For example, if g_1 is described by a mask of the form $\begin{matrix} 0 & 1 \\ -1 & 0 \end{matrix}$ then the directional derivative g_1 at the point (x, y) in the image is

$$(0 \times I(x, y)) + (1 \times I(x+1, y)) + (-1 \times I(x, y+1)) + (0 \times I(x+1, y+1))$$

where $I(i, j)$ is the image intensity at the pixel (i, j).

If, on the other hand, we fit a quadratic surface in a 3×3 window (instead of a planar surface in a 2×2 window) and differentiate the fitted surface to obtain the gradient, we arrive at the discrete masks shown in Figure 2 and known as the *Prewitt operator*.

The operators described above accentuate image intensity variations, thus detecting edges. Moreover, they also accentuate image noise, and thus detect spurious edges. To account for this, the opposite effect of image smoothing should be employed. Smoothing masks have only positive weights. The masks of the *Sobel operator* are designed following this principle; they are basically equivalent to Roberts operators applied after the image has been smoothed. It can be shown that the Sobel operator is the result of the (discrete) *convolution* of an averaging mask with the horizontal and vertical directional derivative masks (Figure 3).

The convolution, denoted by $*$, is the operation of computing the weighted integral (or sum in the case of discrete functions) of one function with another function that has first been reflected about the origin and suitably displaced. If one of the functions is symmetric about the origin, as in the case of the Gaussian, then reflection is unnecessary. The two

$$g_1$$

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$g_2$$

$$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Figure 3: **Sobel operator**: Its application amounts to smoothing followed by differencing.

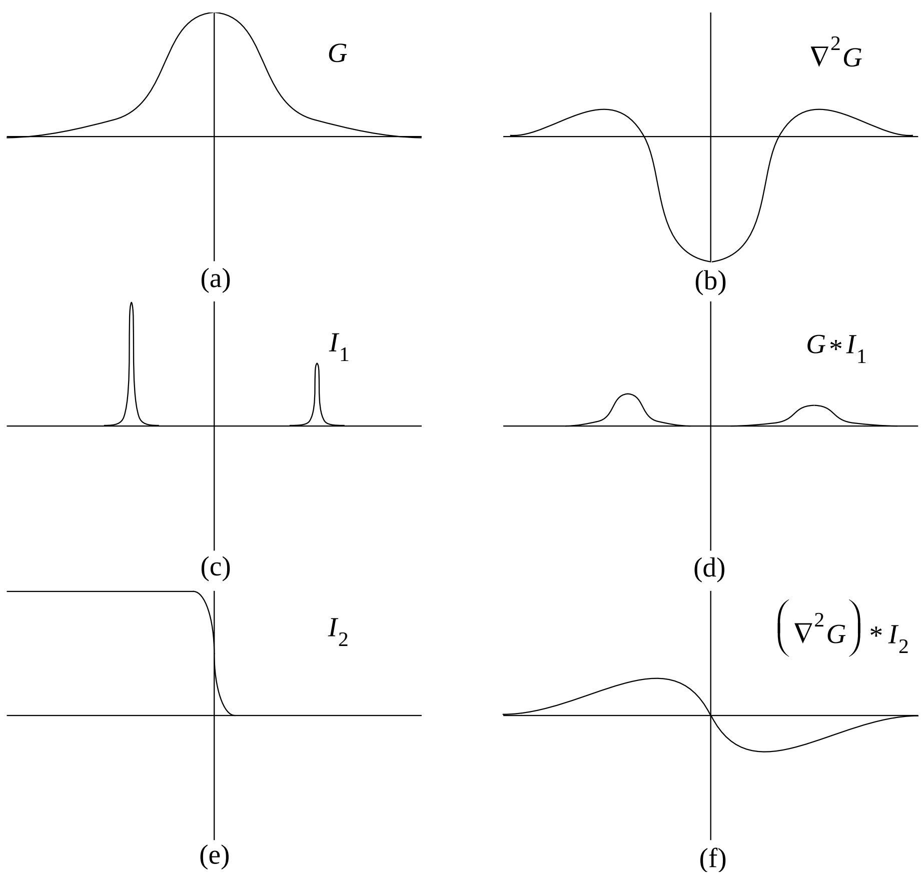

Figure 4: Examples illustrating continuous convolution: (a) shows a one-dimensional Gaussian function G, (b) shows the Laplacian of the Gaussian $\bigtriangledown^2 G$, (c) shows a function I_1 representing two noise spikes in an otherwise zero intensity one-dimensional slice of an image, (d) shows I_1 smoothed by convolving it with G thereby reducing the effect of the noise, (e) shows a function I_2 representing an abrupt change in intensity in a one-dimensional slice of an image, and (f) shows I_2 convolved with the Laplacian of the Gaussian $\bigtriangledown^2 G$ where the zero crossing marks the abrupt change in intensity.

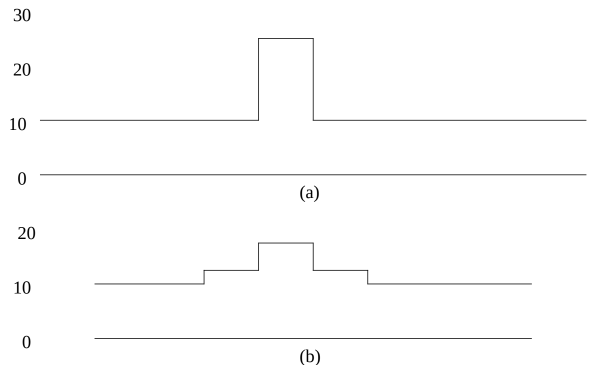

Figure 5: (a) The intensities for a one-dimensional image slice. (b) The result of convolving the image slice with a discrete Gaussian.

functions, say f and g, are said to be *convolved* and the resulting convolution is denoted $f * g$. In the case of two continuous functions f and g of a single variable, $f(x) * g(x) = (f * g)(x) = \int f(x-t)g(t)dt$. Figure 4 provides examples illustrating the result of convolving images with functions such as the Gaussian and the Laplacian of the Gaussian.

In digital image processing, the discrete Gaussian is used to assign to each pixel the weighted average of the intensity of that pixel and the intensities of its neighboring pixels. In the one-dimensional case, the discrete Gaussian $G(i)$ assigns to each integer a positive real number such that $\sum_{-\infty}^{\infty} G(i) = 1.0$. For practical purposes, the weights for integers much greater than 3 or much less than -3 are zero. Here is a simple example of a discrete Gaussian.

$$G(i) = \begin{cases} 0.2 & \text{if } i = -1 \\ 0.6 & \text{if } i = 0 \\ 0.2 & \text{if } i = 1 \\ 0.0 & \text{otherwise} \end{cases}$$

Suppose we have a one-dimensional image slice corresponding to the following vector of image intensities $\langle 10, 10, 10, 10, 25, 10, 10, 10, 10, 10 \rangle$, such that the intensity $I(j)$ at the jth pixel is the jth element of the vector, *e.g.,* $I(5) = 25$ and $I(6) = 10$. Convolving the image slice with the discrete Gaussian amounts to computing the following sum for each pixel except those at the boundaries of the image,

$$I'(j) = \sum_{-\infty}^{\infty} I(i)G(j-i)$$

to obtain the smoothed image slice $I'(j)$ (see Figure 5) corresponding to the vector $\langle 10, 10, 13, 19, 13, 10, 10, 10 \rangle$.

In the case of a two-dimensional image, smoothing involves a two-dimensional Gaussian function $G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ for an appropriately chosen standard deviation σ. A popular edge detector based on Gaussian smoothing is due to Marr and Hildreth [1980]. The Marr-Hildreth operator convolves the image with the Laplacian of a Gaussian function and then takes the zero-crossings of the result to identify edges. In particular, if $\bigtriangledown^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the

(a)

(b) (c)

Figure 6: (a) Original image. (b) Extracted edges at a fine scale. (c) Extracted edges at a coarse scale. (Reproduced from [Canny, 1986] ©1986 IEEE.)

Laplacian operator, G is a Gaussian and $I(x, y)$ the image intensity function, their scheme computes:

$$\bigtriangledown^2(G * I) = (\bigtriangledown^2 G) * I$$

This operator computes the second-order partial derivatives along two orthogonal axes that can be oriented in any fashion since the operator $\bigtriangledown^2 G$ is rotationally symmetric. Along a straight step edge, we can orient one axis along the edge and the other axis perpendicular to the edge. Both partial derivatives are zero in both directions, but the derivative across the edge becomes nonzero as we move away from the edge. Thus, the operator $\bigtriangledown^2 G$ produces zero-crossings along the edge. Figure 6 contains results from the application of the Canny edge detector which uses several one-dimensional edge-segment detectors at each point in the Gaussian-convolved image as a substitute for the Laplacian.