Sobel operator

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.48}$$

The Sobel operator is often used as a simple detector of horizontality and verticality of edges, in which case only masks h_1 and h_3 are used. If the h_1 response is y and the h_3 response x, we might then derive edge strength (magnitude) as

$$\sqrt{x^2 + y^2} \quad \text{or} \quad |x| + |y| \tag{4.49}$$

and direction as $\tan^{-1}(y/x)$.

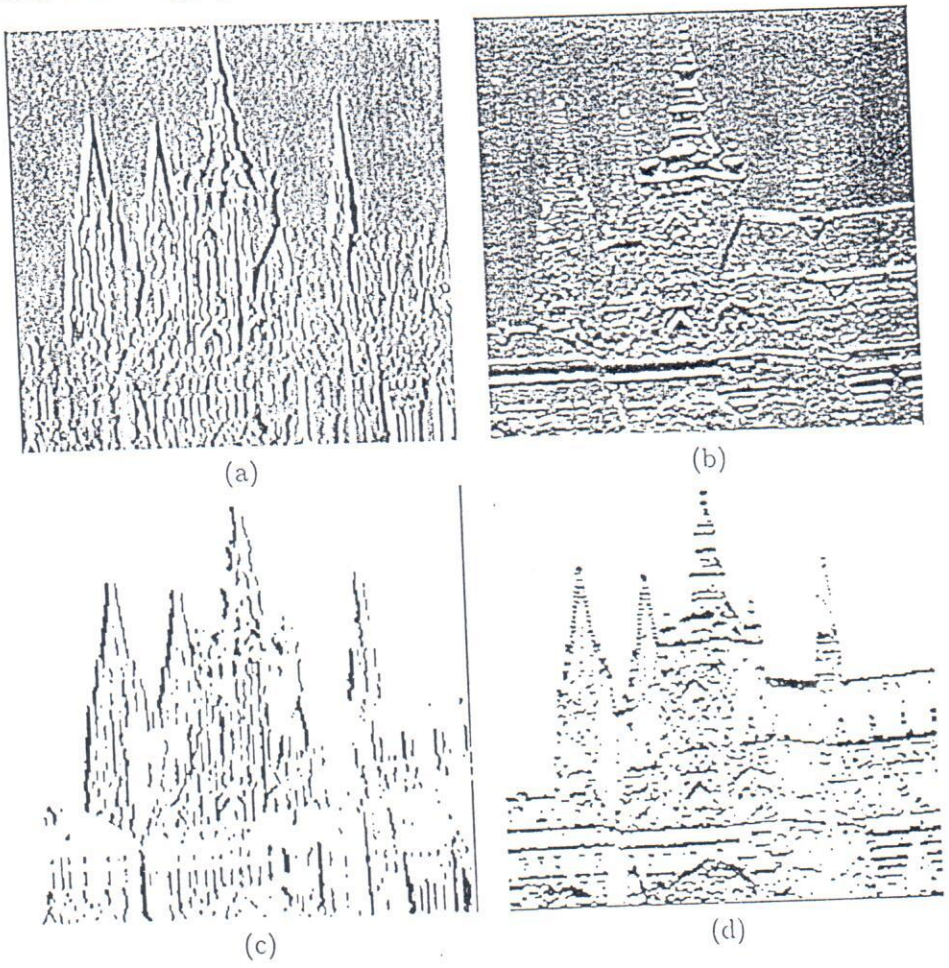

Figure 4.19: *First-derivative edge detection using Prewitt compass operators: (a) north direction (the brighter the pixel value, the stronger the edge); (b) east direction; (c) strong edges from (a); (d) strong edges from (b).*

Robinson operator

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix} \tag{4.50}$$

Kirsch operator

$$h_1 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix} \quad h_2 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix} \quad h_3 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix} \tag{4.51}$$

To illustrate the application of gradient operators on real images, consider again the image given in Figure 4.10a. The Laplace edge image calculated is shown in Figure 4.18a; the value of the operator has been histogram equalized to enhance its visibility.

The properties of an operator approximating the first derivative are demonstrated using the Prewitt operator—results of others are similar. The original image is again given in Figure 4.10a; Prewitt approximations to the directional gradients are in Figures 4.19a,b, in which north and east directions are shown. Significant edges (those with above-threshold magnitude) in the two directions are given in Figures 4.19c,d.

4.3.3 Zero-crossings of the second derivative

In the 1970s, Marr's theory (see Section 9.1.1) concluded from neurophysiological experiments that object boundaries are the most important cues that link an intensity image with its interpretation. Edge detection techniques existing at that time (e.g., the Kirsch, Sobel, and Pratt operators) were based on convolution in very small neighborhoods and worked well only for specific images. The main disadvantage of these edge detectors is their dependence on the size of the object and sensitivity to noise.

An edge detection technique based on the **zero-crossings** of the second derivative (in its original form, the **Marr-Hildreth** edge detector [Marr and Hildreth 80] or the same paper in a more recent collection, [Marr and Hildreth 91]) explores the fact that a step edge corresponds to an abrupt change in the image function. The first derivative of the image function should have an extremum at the position corresponding to the edge in the image, and so the second derivative should be zero at the same position; however, it is much easier and more precise to find a zero-crossing position than an extremum. In Figure 4.20 this principle is illustrated in 1D for the sake of simplicity. Figure 4.20a shows step edge profiles of the original image function with two different slopes, Figure 4.20b depicts the first derivative of the image function, and Figure 4.20c illustrates the second derivative; notice that this crosses the zero level at the same position as the edge. Considering a step-like edge in 2D, the 1D profile of Figure 4.20a corresponds to a cross section through the 2D step. The steepness of the profile will change if the orientation of the cutting plane changes—the maximum steepness is observed when the plane is perpendicular to the edge direction.

The crucial question is how to compute the second derivative robustly. One possibility is to smooth an image first (to reduce noise) and then compute second derivatives. When choosing a smoothing filter, there are two criteria that should be fulfilled [Marr and Hildreth 80]. First, the filter should be smooth and roughly band limited in the frequency domain to

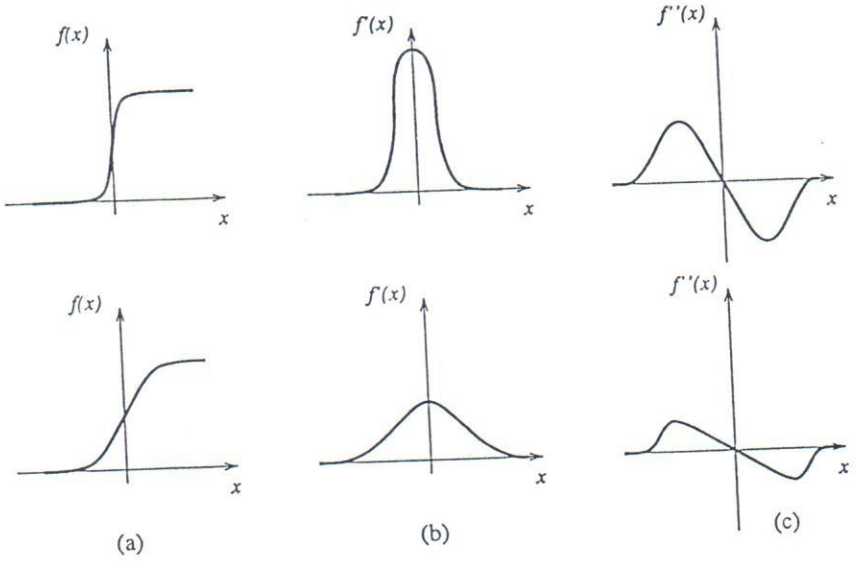

Figure 4.20: *1D edge profile of the zero-crossing.*

reduce the possible number of frequencies at which function changes can take place. Second, the constraint of spatial localization requires the response of a filter to be from nearby points in the image. These two criteria are conflicting, but they can be optimized simultaneously using a Gaussian distribution. In practice, one has to be more precise about what is meant by the localization performance of an operator, and the Gaussian may turn out to be sub-optimal. We shall consider this in the next section.

The 2D Gaussian smoothing operator $G(x, y)$ (also called a Gaussian filter, or simply a Gaussian) is given by

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.52}$$

where x, y are the image co-ordinates and σ is a standard deviation of the associated probability distribution. Sometimes this is presented with a normalizing factor:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad \text{or} \quad G(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The standard deviation σ is the only parameter of the Gaussian filter—it is proportional to the size of the neighborhood on which the filter operates. Pixels more distant from the center of the operator have smaller influence, and pixels farther than 3σ from the center have negligible influence.

Our goal is to obtain a second derivative of a smoothed 2D function $f(x, y)$. We have already seen that the Laplace operator ∇^2 gives the second derivative, and is non-directional (isotropic). Consider then the Laplacian of an image $f(x, y)$ smoothed by a Gaussian (expressed using a convolution $*$). The operation is abbreviated by some authors as **LoG**, from **Laplacian of Gaussian**.

$$\nabla^2[G(x, y, \sigma) * f(x, y)] \tag{4.53}$$

The order of performing differentiation and convolution can be interchanged because of the linearity of the operators involved:

$$[\nabla^2 G(x, y, \sigma)] * f(x, y) \tag{4.54}$$

The derivative of the Gaussian filter $\nabla^2 G$ can be pre-computed analytically, since it is independent of the image under consideration. Thus, the complexity of the composite operation is reduced. For simplicity, we use the substitution $r^2 = x^2 + y^2$, where r measures distance from the origin; this is reasonable, as the Gaussian is circularly symmetric. This substitution converts the 2D Gaussian [equation (4.52)] into a 1D function that is easier to differentiate:

$$G(r) = e^{-\frac{r^2}{2\sigma^2}} \tag{4.55}$$

The first derivative $G'(r)$ is then

$$G'(r) = -\frac{1}{\sigma^2}\, r\, e^{-\frac{r^2}{2\sigma^2}} \tag{4.56}$$

and the second derivative $G''(r)$, the Laplacian of a Gaussian, is

$$G''(r) = \frac{1}{\sigma^2}\left(\frac{r^2}{\sigma^2} - 1\right) e^{-\frac{r^2}{2\sigma^2}} \tag{4.57}$$

After returning to the original co-ordinates x, y and introducing a normalizing multiplicative coefficient c, we get a convolution mask of a LoG operator:

$$h(x, y) = c\left(\frac{x^2 + y^2 - \sigma^2}{\sigma^4}\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.58}$$

where c normalizes the sum of mask elements to zero. Because of its shape, the inverted LoG operator is commonly called a **Mexican hat**. Examples of discrete approximations of 5×5 and 17×17 LoG operators $\nabla^2 G$ follow [Jain et al. 95]:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & -1 & 0 & 0 \\
0 & 0 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & 0 & 0 \\
0 & -1 & -1 & -2 & -3 & -3 & -3 & -2 & -3 & -2 & -3 & -3 & -3 & -2 & -1 & -1 & 0 \\
0 & -1 & -2 & -3 & -3 & -3 & 0 & 2 & 4 & 2 & 0 & -3 & -3 & -3 & -2 & -1 & 0 \\
-1 & -1 & -3 & -3 & -3 & 0 & 4 & 10 & 12 & 10 & 4 & 0 & -3 & -3 & -3 & -1 & -1 \\
-1 & -1 & -3 & -3 & -2 & 2 & 10 & 18 & 21 & 18 & 10 & 2 & -2 & -3 & -3 & -1 & -1 \\
-1 & -1 & -3 & -3 & -3 & 4 & 12 & 21 & 24 & 21 & 12 & 4 & -3 & -3 & -3 & -1 & -1 \\
-1 & -1 & -3 & -3 & -2 & 2 & 10 & 18 & 21 & 18 & 10 & 2 & -2 & -3 & -3 & -1 & -1 \\
-1 & -1 & -3 & -3 & -3 & 0 & 4 & 10 & 12 & 10 & 4 & 0 & -3 & -3 & -3 & -1 & -1 \\
0 & -1 & -2 & -3 & -3 & -3 & 0 & 2 & 4 & 2 & 0 & -3 & -3 & -3 & -2 & -1 & 0 \\
0 & -1 & -1 & -2 & -3 & -3 & -3 & -2 & -3 & -2 & -3 & -3 & -3 & -2 & -1 & -1 & 0 \\
0 & 0 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & 0 & 0 \\
0 & 0 & -1 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

Finding second derivatives in this way is very robust. Gaussian smoothing effectively suppresses the influence of the pixels that are up to a distance 3σ from the current pixel; then the Laplace operator is an efficient and stable measure of changes in the image.

After the image convolution with $\nabla^2 G$, the locations in the convolved image where the zero level is crossed correspond to the positions of edges. The advantage of this approach compared to classical edge operators of small size is that a larger area surrounding the current pixel is taken into account; the influence of more distant points decreases according to the σ of the Gaussian. In the ideal case of an isolated step edge, the σ variation does not affect the location of the zero-crossing.

Convolution masks become large for larger σ; for example, $\sigma = 4$ needs a mask about 40 pixels wide. Fortunately, there is a separable decomposition of the $\nabla^2 G$ operator [Huertas and Medioni 86] that can speed up computation considerably.

The practical implication of Gaussian smoothing is that edges are found reliably. If only globally significant edges are required, the standard deviation σ of the Gaussian smoothing filter may be increased, having the effect of suppressing less significant evidence.

The $\nabla^2 G$ operator can be very effectively approximated by convolution with a mask that is the difference of two Gaussian averaging masks with substantially different σ—this method is called the **difference of Gaussians**, abbreviated as **DoG**. The correct ratio of the standard deviations σ of the Gaussian filters is discussed in [Marr 82].

Even coarser approximations to $\nabla^2 G$ are sometimes used—the image is filtered twice by an averaging operator with smoothing masks of different sizes.

When implementing a zero-crossing edge detector, trying to detect *zeros* in the LoG or DoG image will inevitably fail, while naive approaches of thresholding the LoG/DoG image and defining the zero-crossings in some interval of values close to zero give piecewise disconnected edges at best. To end up with a well-functioning second-derivative edge detector, it is necessary to implement a true zero-crossing detector. A simple detector may identify a zero crossing in a moving 2×2 window, assigning an edge label to any one corner pixel, say

the upper left, if LoG/DoG image values of both polarities occur in the 2×2 window; no edge label would be given if values within the window are either all positive or all negative. Another post-processing step to avoid detection of zero-crossings corresponding to nonsignificant edges in regions of almost constant gray-level would admit only those zero-crossings for which there is sufficient edge evidence from a first-derivative edge detector. Figure 4.21 provides several examples of edge detection using zero crossings of the second derivative.

Many other approaches improving zero-crossing performance can be found in the literature [Qian and Huang 94, Mehrotra and Shiming 96]; some of them are used in pre-processing [Hardie and Boncelet 95] or post-processing steps [Alparone et al. 96].

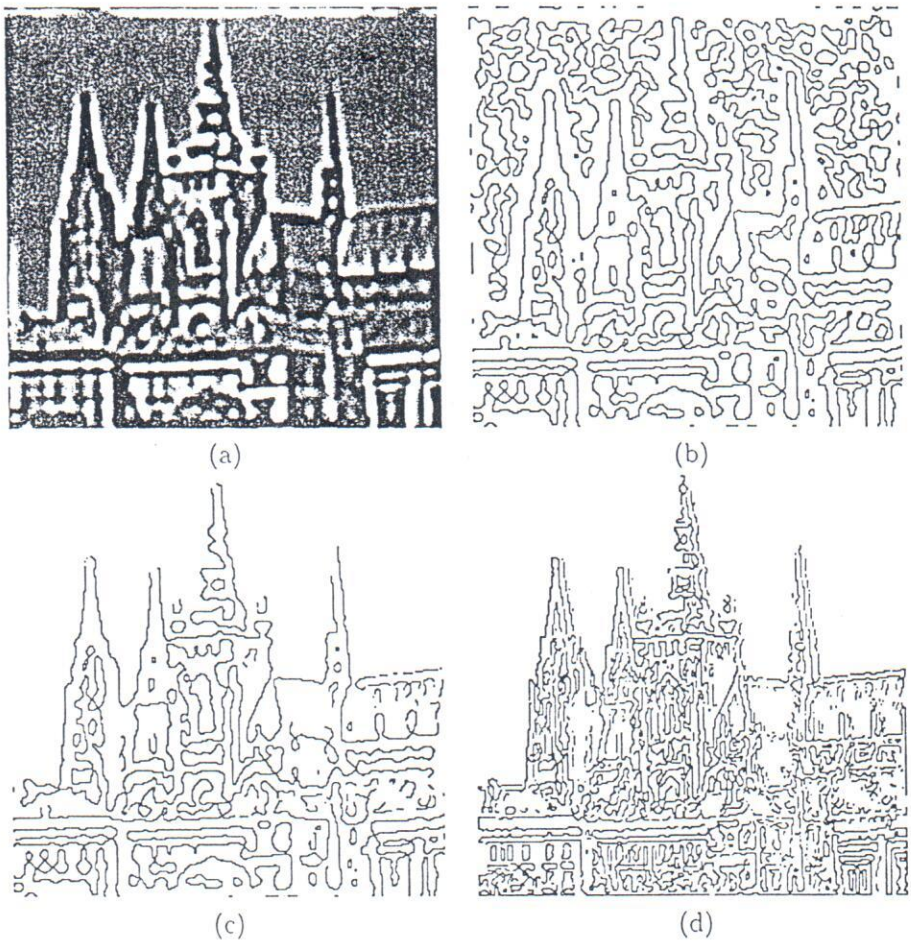

Figure 4.21: *Zero-crossings of the second derivative, see Figure 4.10a for the original image: (a) DoG image ($\sigma_1 = 0.10, \sigma_2 = 0.09$), dark pixels correspond to negative DoG values, bright pixels represent positive DoG values; (b) zero-crossings of the DoG image; (c) DoG zero-crossing edges after removing edges lacking first-derivative support; (d) LoG zero-crossing edges ($\sigma = 0.20$) after removing edges lacking first-derivative support—note different scale of edges due to different Gaussian smoothing parameters.*

The traditional second-derivative zero-crossing technique has disadvantages as well. First, it smoothes the shape too much; for example, sharp corners are lost. Second, it tends to create closed loops of edges (nicknamed the 'plate of spaghetti' effect). Although this property was highlighted as an advantage in original papers, it has been seen as a drawback in many applications.

Neurophysiological experiments [Marr 82, Ullman 81] provide evidence that the human eye retina in the form of the **ganglion cells** performs operations very similar to the $\nabla^2 G$ operations. Each such cell responds to light stimuli in a local neighborhood called the **receptive field**, which has a center-surround organization of two complementary types, off-center and on-center. When a light stimulus occurs, activity of on-center cells increases and that of off-center cells is inhibited. The retinal operation on the image can be described analytically as the convolution of the image with the $\nabla^2 G$ operator.

4.3.4 Scale in image processing

Many image processing techniques work locally, theoretically at the level of individual pixels—edge detection methods are an example. The essential problem in such computation is **scale**. Edges correspond to the gradient of the image function, which is computed as a difference between pixels in some neighborhood. There is seldom a sound reason for choosing a particular size of neighborhood, since the 'right' size depends on the size of the objects under investigation. To know what the objects are assumes that it is clear how to interpret an image, and this is not in general known at the pre-processing stage. The solution to the problem formulated above is a special case of a general paradigm called the **system approach**. This methodology is common in cybernetics or general system theory to study complex phenomena.

The phenomenon under investigation is expressed at different resolutions of the description, and a formal model is created at each resolution. Then the qualitative behavior of the model is studied under changing resolution of the description. Such a methodology enables the deduction of meta-knowledge about the phenomenon that is not seen at the individual description levels.

Different description levels are easily interpreted as different scales in the domain of digital images. The idea of scale is fundamental to Marr's edge detection technique, introduced in Section 4.3.3, where different scales are provided by different sizes of Gaussian filter masks. The aim was not only to eliminate fine scale noise but also to separate events at different scales arising from distinct physical processes [Marr 82].

Assume that a signal has been smoothed with several masks of variable sizes. Every setting of the scale parameters implies a different description, but it is not known which one is correct; for many tasks, no one scale is categorically correct. If the ambiguity introduced by the scale is inescapable, the goal of scale-independent description is to reduce this ambiguity as much as possible.

Many publications tackle scale-space problems, e.g., [Hummel and Moniot 89, Perona and Malik 90, Williams and Shah 90, Mokhtarian and Mackworth 92, Mokhtarian 95, Morrone et al. 95, Elder and Zucker 96, Aydin et al. 96, Lindeberg 96]. A symbolic approach to constructing a multi-scale primitive shape description to 2D binary (contour) shape images is presented in [Saund 90], and the use of a scale-space approach for object recognition is in [Topkar et al. 90]. Here we shall consider just three examples of the application of multiple

scale description to image analysis.

The first approach [Lowe 89] aims to process planar noisy curves at a range of scales—the segment of curve that represents the underlying structure of the scene needs to be found. The problem is illustrated by an example of two noisy curves; see Figure 4.22. One of these may be interpreted as a closed (perhaps circular) curve, while the other could be described as two intersecting straight lines. Local tangent direction and curvature of the curve are significant

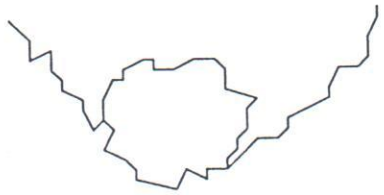

Figure 4.22: *Curves that may be analyzed at multiple scales.*

only with some idea of scale after the curve is smoothed by a Gaussian filter with varying standard deviations.

A second approach [Witkin 83], called **scale-space filtering**, tries to describe signals qualitatively with respect to scale. The problem was formulated for 1D signals $f(x)$, but it can easily be generalized for 2D functions as images. The original 1D signal $f(x)$ is smoothed by convolution with a 1D Gaussian,

$$G(x,\sigma) = e^{-\frac{x^2}{2\sigma^2}} \tag{4.59}$$

If the standard deviation σ is slowly changed, the function

$$F(x,\sigma) = f(x) * G(x,\sigma) \tag{4.60}$$

represents a surface on the (x,σ) plane that is called the **scale-space image**. Inflection points of the curve $F(x,\sigma_0)$ for a distinct value σ_0,

$$\frac{\partial^2 F(x,\sigma_0)}{\partial x^2} = 0 \quad \frac{\partial^3 F(x,\sigma_0)}{\partial x^3} \neq 0 \tag{4.61}$$

describe the curve $f(x)$ qualitatively. The positions of inflection points can be drawn as a set of curves in (x,σ) co-ordinates (see Figure 6.15). Coarse to fine analysis of the curves corresponding to inflection points, i.e., in the direction of decreasing value of the σ, localizes large-scale events.

The qualitative information contained in the scale-space image can be transformed into a simple **interval tree** that expresses the structure of the signal $f(x)$ over all observed scales. The interval tree is built from the root that corresponds to the largest scale ($\sigma_{\max}$), and then the scale-space image is searched in the direction of decreasing σ. The interval tree branches at those points where new curves corresponding to inflection points appear (see Chapter 6 and Section 6.2.4).

The third example of the application of scale is that used by the popular **Canny edge detector**. Since the Canny detector is a significant and widely used contribution to edge detection techniques, its principles will be explained in detail.

4.3.5 Canny edge detection

Canny proposed a new approach to edge detection [Canny 83, Brady 84, Canny 86] that is optimal for step edges corrupted by white noise. The optimality of the detector is related to three criteria.

- The **detection** criterion expresses the fact that important edges should not be missed and that there should be no spurious responses.
- The **localization** criterion says that the distance between the actual and located position of the edge should be minimal.
- The **one response** criterion minimizes multiple responses to a single edge. This is partly covered by the first criterion, since when there are two responses to a single edge, one of them should be considered as false. This third criterion solves the problem of an edge corrupted by noise and works against non-smooth edge operators [Rosenfeld and Thurston 71].

Canny's derivation of a new edge detector is based on several ideas.

1. The edge detector was expressed for a 1D signal and the first two optimality criteria. A closed-form solution was found using the calculus of variations.
2. If the third criterion (multiple responses) is added, the best solution may be found by numerical optimization. The resulting filter can be approximated effectively with error less than 20% by the first derivative of a Gaussian smoothing filter with standard deviation σ [Canny 86]; the reason for doing this is the existence of an effective implementation. There is a strong similarity here to the Marr-Hildreth edge detector [Marr and Hildreth 80], which is based on the Laplacian of a Gaussian—see Section 4.3.3.
3. The detector is then generalized to two dimensions. A step edge is given by its position, orientation, and possibly magnitude (strength). It can be shown that convolving an image with a symmetric 2D Gaussian and then differentiating in the direction of the gradient (perpendicular to the edge direction) forms a simple and effective directional operator (recall that the Marr-Hildreth zero-crossing operator does not give information about edge direction, as it uses a Laplacian filter).

   Suppose G is a 2D Gaussian [equation (4.52)] and assume we wish to convolve the image with an operator G_n which is a first derivative of G in the direction $\mathbf{n}$.

$$G_n = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla G \tag{4.62}$$

   The direction $\mathbf{n}$ should be oriented perpendicular to the edge. Although this direction is not known in advance, a robust estimate of it based on the smoothed gradient direction is available. If f is the image, the normal to the edge $\mathbf{n}$ is estimated as

$$\mathbf{n} = \frac{\nabla(G * f)}{|\nabla(G * f)|} \tag{4.63}$$

The edge location is then at the local maximum of the image f convolved with the operator G_n in the direction $\mathbf{n}$.

$$\frac{\partial}{\partial \mathrm{n}} G_n * f = 0 \tag{4.64}$$

Substituting in equation (4.64) for G_n from equation (4.62), we get

$$\frac{\partial^2}{\partial \mathrm{n}^2} G * f = 0 \tag{4.65}$$

This equation (4.65) illustrates how to find local maxima in the direction perpendicular to the edge; this operation is often referred to as **non-maximal suppression** (see also Algorithm 5.5).

As the convolution and derivative are associative operations in equation (4.65), we can first convolve an image f with a symmetric Gaussian G and then compute the directional second-derivative using an estimate of the direction $\mathbf{n}$ computed according to equation (4.63). The strength of the edge (magnitude of the gradient of the image intensity function f) is measured as

$$|G_n * f| = |\nabla(G * f)| \tag{4.66}$$

A different generalization of this optimal detector into two dimensions was proposed by Spacek [Spacek 86], and the problem of edge localization is revisited in [Tagare and deFigueiredo 90].

4. Spurious responses to the single edge caused by noise usually create a 'streaking' problem that is very common in edge detection in general. The output of an edge detector is usually thresholded to decide which edges are significant, and streaking means the breaking up of the edge contour caused by the operator fluctuating above and below the threshold. Streaking can be eliminated by **thresholding with hysteresis**. If any edge response is above a *high threshold*, those pixels constitute definite edge output of the detector for a particular scale. Individual weak responses usually correspond to noise, but if these points are connected to any of the pixels with strong responses, they are more likely to be actual edges in the image. Such connected pixels are treated as edge pixels if their response is above a *low threshold*. The low and high thresholds are set according to an estimated signal-to-noise ratio [Canny 86] (see also Algorithm 5.6).

5. The correct scale for the operator depends on the objects contained in the image. The solution to this unknown is to use multiple scales and aggregate information from them. Different scales for the Canny detector are represented by different standard deviations σ of the Gaussians. There may be several scales of operators that give significant responses to edges (i.e., signal-to-noise ratio above the threshold); in this case the operator with the smallest scale is chosen, as it gives the best localization of the edge.

   Canny proposed a **feature synthesis** approach. All significant edges from the operator with the smallest scale are marked first, and the edges of a hypothetical operator with larger σ are synthesized from them (i.e., a prediction is made of how the large σ should perform on the evidence gleaned from the smaller σ—see also Section 4.3.4 and

Figure 6.15). Then the synthesized edge response is compared with the actual edge response for larger σ. Additional edges are marked only if they have a significantly stronger response than that predicted from synthetic output.

This procedure may be repeated for a sequence of scales, a cumulative edge map being built by adding those edges that were not identified at smaller scales.

Algorithm 4.4: Canny edge detector

1. Convolve an image f with a Gaussian of scale σ.
2. Estimate local edge normal directions $\mathbf{n}$ using equation (4.63) for each pixel in the image.
3. Find the location of the edges using equation (4.65) (non-maximal suppression).
4. Compute the magnitude of the edge using equation (4.66).
5. Threshold edges in the image with hysteresis to eliminate spurious responses.
6. Repeat steps (1) through (5) for ascending values of the standard deviation σ.
7. Aggregate the final information about edges at multiple scale using the 'feature synthesis' approach.

(a) (b)

Figure 4.23: *Canny edge detection at two different scales.*

Figure 4.23a shows the edges of Figure 4.10a detected by a Canny operator with $\sigma = 1.0$. Figure 4.23b shows the edge detector response for $\sigma = 2.8$ (feature synthesis has not been applied here).