

Figure 5.22: Curves that may be analyzed at multiple scales.
© Cengage Learning 2015.

$f(x)$ is smoothed by convolution with a 1D Gaussian

$$G(x, \sigma) = e^{-x^2/2\sigma^2}. \quad (5.51)$$

If the standard deviation σ is slowly changed, the function

$$F(x, \sigma) = f(x) * G(x, \sigma) \quad (5.52)$$

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

$$\frac{\partial^2 F(x, \sigma_0)}{\partial x^2} = 0 \quad \text{and} \quad \frac{\partial^3 F(x, \sigma_0)}{\partial x^3} \neq 0 \quad (5.53)$$

describe the curve $f(x)$ qualitatively. The positions of inflexion points can be drawn as a set of curves in (x, σ) co-ordinates (see Figure 8.16). Coarse to fine analysis of the curves corresponding to inflexion 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 inflexion points appear (see Chapter 8 and Section 8.2.4).

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.

5.3.5 Canny edge detection

Canny proposed an approach to edge detection [Canny, 1986] 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, 1971].

Canny's derivation 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, 1986]; the reason for doing this is the existence of an effective implementation. There is a strong similarity here to the LoG based Marr-Hildreth edge detector [Marr and Hildreth, 1980], see Section 5.3.3.
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 (5.47)] and assume we wish to convolve the image with an operator G_n which is a first derivative of G in some direction $\mathbf{n}$

$$G_n = \frac{\partial G}{\partial \mathbf{n}} = \mathbf{n} \nabla G. \quad (5.54)$$

We would like $\mathbf{n}$ to be perpendicular to the edge: this direction is not known in advance, but 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

$$\mathbf{n} = \frac{\nabla(G * f)}{|\nabla(G * f)|}. \quad (5.55)$$

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

$$\frac{\partial}{\partial \mathbf{n}} G_n * f = 0. \quad (5.56)$$

Substituting in equation (5.56) for G_n from equation (5.54), we get

$$\frac{\partial^2}{\partial \mathbf{n}^2} G * f = 0. \quad (5.57)$$

This equation (5.57) 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 6.4).

As the convolution and derivative are associative operations in equation (5.57), 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 (5.55). The strength of the edge (magnitude of the gradient of the image intensity function f) is measured as

$$|G_n * f| = |\nabla(G * f)|. \quad (5.58)$$

4. Spurious responses to a 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 may break up edge contours as the operator fluctuates above and below the threshold. Streaking can be eliminated by **thresholding with hysteresis**, employing a hard (high) threshold and a soft (lower) threshold—see Algorithm 6.5. The low and high thresholds are set according to an estimated signal-to-noise ratio [Canny, 1986].
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 5.3.4 and Figure 8.16). 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 5.4: Canny edge detector

1. Convolve an image f with a Gaussian of scale σ .
2. Estimate local edge normal directions $\mathbf{n}$ using equation (5.55) for each pixel in the image.
3. Find the location of the edges using equation (5.57) (non-maximal suppression).
4. Compute the magnitude of the edge using equation (5.58).
5. Threshold edges in the image with hysteresis (Algorithm 6.5) 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.

Figure 5.23a shows the edges of Figure 5.9a detected by a Canny operator with $\sigma = 1.0$. Figure 5.23b shows the edge detector response for $\sigma = 2.8$ (feature synthesis has not been applied here).

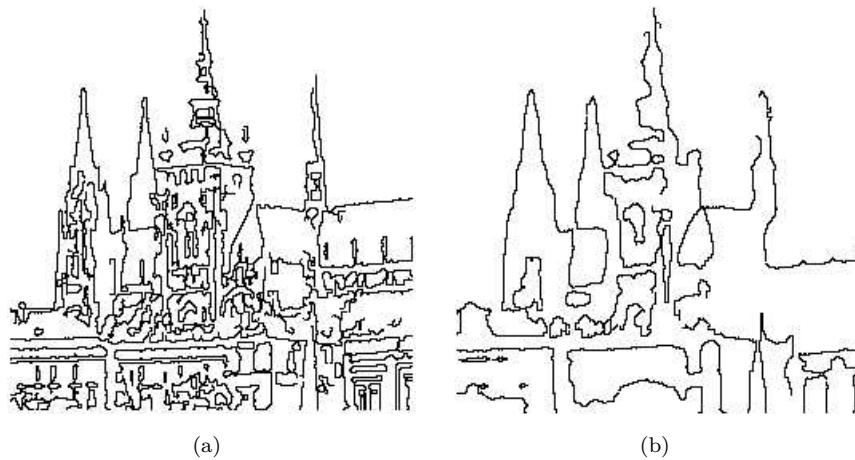


Figure 5.23: Canny edge detection at two different scales. © Cengage Learning 2015.

Canny's detector represents a complicated but major contribution to edge detection. Its full implementation is unusual, it being common to find implementations that omit feature synthesis—that is, just steps 1–5 of Algorithm 5.4.

5.3.6 Parametric edge models

Parametric models are based on the idea that the discrete image intensity function can be considered a sampled and noisy approximation of an underlying continuous or piecewise continuous image intensity function [Nevatia, 1977]. While this function is not known, it can be estimated from the available discrete image intensity function and image properties can be determined from this continuous estimate, possibly with subpixel precision. It is usually impossible to represent image intensities using a single continuous function since a single function leads to high-order intensity functions in x and y . Instead, piecewise continuous function estimates called **facets** are used to represent (a neighborhood of) each image pixel. Such an image representation is called a **facet model** [Haralick and Watson, 1981; Haralick, 1984; Haralick and Shapiro, 1992].

The intensity function in a neighborhood can be estimated using models of different complexity. The simplest one is the flat facet model that uses piecewise constants and each pixel neighborhood is represented by a flat function of constant intensity. The sloped model uses piecewise linear functions forming a sloped plane fitted to local image intensities. Quadratic and bi-cubic facet models employ correspondingly more complex functions.

Once the facet model parameters are available for each image pixel, edges can be detected as extrema of the first directional derivative and/or zero-crossings of the second directional derivative of the local continuous facet model functions.

A thorough treatment of facet models and their modifications for peak noise removal, segmentation into constant-gray-level regions, determination of statistically significant edges, gradient edge detection, directional second-derivative zero-crossing edge detection, and line and corner detection is given in [Haralick and Shapiro, 1992]. Importantly, techniques for facet model parameter estimation are given there.