FEATURE EXTRACTION

Images are the result of a number of complex interactions between the objects in the scene being observed, the light coming from the different sources, and the imaging apparatus itself. While we understand these interactions, and can use them to generate photorealistic synthetic images, the inverse problem, which consists of understanding the semantic contents of a scene, is quite difficult. Since the images are the result of a projection from 3-D to 2-D, it is necessary to invoke generic constraints, such as smoothness, to infer the structure of the scene. The accepted methodology is to proceed in a hierarchical manner, processing small neighborhoods of the images to generate partial descriptions, and aggregating these into more global ones. The extraction of such features is the topic of this article. Feature extraction therefore involves the inference of primitives directly from the image, or from partial descriptions of it. For ease of presentation, we will only consider two levels of this hierarchy, and describe the extraction of features directly from the images, then discuss the extraction of higher level primitives as a representation issue.

Feature extraction is an essential component of any image analysis or understanding task. Whether we want to match two or more images (to establish depth in stereo processing, or compute motion in an image sequence), or match an image with a model (to establish the presence and find the pose and/or the identity of an object in a scene, or determine changes), reasoning directly at the level of the raw image is rarely appropriate, and we need to abstract features from the data. Clearly, the definition of a good feature depends on the task at hand, and on the expected variations of the environment:

- *Illumination.* if the lighting is controlled, the raw intensity values may be directly useful.
- *Viewpoint.* if the viewpoint is fixed, parts of the scene may be ignored.

- *Range of observation.* for inspection problems, a piece may always be presented with the same aspect, or rotated in a plane only, or only appear within a range of scales. This analysis leads to different criteria for a good feature.
- *Segmentation.* if the object(s) of interest can be easily separated from the background, for instance dark flat objects on a light table, the contours of the object(s) are excellent features.
- *Occlusion.* if an object only appears individually and can be segmented, global features of its appearance are good features for analysis.

We can therefore define the following criteria for good features:

1. *Distinctness.* a feature should reflect that the property it captures is different from its neighbors. Any external property should satisfy this criterion (local maximum of intensity, curvature, . . .)
2. *Invariance.* the presence, position, and properties of a feature should be invariant (or slowly changing) with respect to the expected variations of the observation parameters, such as lighting, and imaging distortions.
3. *Stability.* the detection and properties of a feature should vary smoothly with respect to variations in viewpoint. Under perspective projection, the edge between two faces of a polyhedron is a good feature, whereas the length of this edge is not.

We now discuss in detail the steps involved in extracting and representing features. We start with the extraction of point features from images, then describe the extraction and representation of curve features, then the extraction and representation of region features, and turn our attention to methods which aim at deriving integrated descriptions in terms of multiple features.

EXTRACTION OF POINT FEATURES

Extraction Directly from Gray Level Images

It was observed early that uniform areas in an image do not provide much local information, and thus not suited for template matching or correlation procedures. Authors instead proposed to use an *interest operator,* or *cornerness operator,* to isolate areas with high local variance. One of the first procedures to achieve this goal was the Moravec interest operator (1), as described next.

At each pixel i,j of the image, we compute the variance in a square window of size $2a + 1$ centered around the pixel,

$$\mathrm{var}(i,j) = \frac{\sum_{-a<k,l<a}[I(i,j) - I(i+k, j+l)]l}{2}$$

We then define, for each pixel, the interest operator value as the minimum of the variance values in a small neighborhood. Next, we check whether the value is minimum, again by comparing with values in the same neighborhood. Finally, candidate points are selected by thresholding with respect to a fixed value, chosen empirically to produce a fraction of the image points.

While this procedure is simple and straightforward to implement, it is somewhat ad hoc and suffers from a number of shortcomings, including instability in localization.

A better formulation was proposed by Beaudet (2), who introduced the rotationally invariant operator det by considering a second-order Taylor's expansion of the intensity surface $I(x, y)$:

$$\det(x,y) = I_{xx}I_{yy} - I_{xy}^2 \tag{1}$$

Corners are then defined as local maxima of the absolute value of this measure.

By considering the image $I(x, y)$ as a surface, it is possible to understand this approach with tools of differential geometry:

The previous measure, det is nothing but the determinant of the Hessian matrix,

$$H = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix}$$

It is therefore related to the Gaussian curvature, which is the product of the two principal curvatures $k_{\min}$ and $k_{\max}$ as follows

$$k_{\min} \times k_{\max} = \frac{\det}{(1 + I_x^2 + I_y^2)^2}$$

Note that this operator ignores the finer classification of the intensity surface as elliptic ($k_{\min}k_{\max} > 0$), hyperbolic ($k_{\min}k_{\max} < 0$) or parabolic ($k_{\min}k_{\max} = 0$).

Kitchen and Rosenfeld (3) proposed instead a measure of cornerness based on the following expression.

$$K = \frac{I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_xI_y}{I_x^2 + I_y^2} \tag{2}$$

It can be shown that this corresponds to the second directional derivative in the direction normal to the gradient.

Whereas the previous measures involve second order operators, Noble (4) investigates first order differentials, such as the Plessey corner detector (5), where

$$C_p = \frac{\mathrm{trace}\,\hat{C}}{\det \hat{C}} \quad \text{and} \quad \hat{C} = \begin{bmatrix} \hat{I}_x^2 & \hat{I}_x I_y \\ I_x \hat{I}_y & \hat{I}_y^2 \end{bmatrix} \tag{3}$$

and $\hat{I}$ represents a smoothed version of I.

In order to obtain a dimensionless measure, one should compute instead

$$C_p = \frac{\mathrm{trace}^2\,\hat{C}}{\det \hat{C}} \quad \text{or} \quad C_p = \det \hat{C} - k\,\mathrm{trace}^2\,\hat{C} \tag{4}$$

As noted by Deriche and Giraudon (6), these approaches allow the detection of corners, but the localization of these corners is erroneous for an L junction, and produce multiple responses for trihedral vertices. They propose to use multiple scales of processing to differentiate between these different junction types, and to refine their localization. They show that the exact position of a corner can be detected as a stable zero-crossing in scale-space, and that Beaudet's local maximum moves in scale space along a line which passes through the true position of the vertex.

They therefore use the points detected by Beaudet's measure at two different scales of smoothing to extrapolate the true location of the vertex, as explained below:

- Compute the Laplacian image
- Compute two det functions using scales σ_1 and σ_2
- Threshold and detect the elliptic maxima A and B in $\det_{\sigma 1}$ and $\det_{\sigma 2}$, with $\sigma_2 > \sigma_1$
- Compute the line joining A to B
- Select the point on the line AB where a zero-crossing occurs in the Laplacian image

Smith and Brady (7) propose to bypass derivative computations in their smallest univalue segment assimilating nucleus (SUSAN) corner detector. It computes the area $n(x, y)$ of points inside a circular region N_{xy} which has a brightness similar to the brightness of the center pixel (x, y):

$$n(x,y) = \sum_{(i,j)\, in\, N_{xy}} e^{-(I_{ij}-I_{xy})^{\frac{2}{t}}} \tag{5}$$

The parameter t controls noise sensitivity, and the value of $n(x, y)$ is compared to the maximum possible value $n_{\max}$ leading to the corner strength function

$$cs(x,y) = \begin{matrix} \dfrac{n_{\max}}{2} - n(x,y) & \text{if } n(x,y) < \dfrac{n_{\max}}{2} \\ 0 & \text{otherwise} \end{matrix} \tag{6}$$

To reduce false positives, two additional criteria must be satisfied:

1. The center of gravity of the circular region must be distant from the center point, and
2. All pixels on the line between the center point and the center of gravity must have similar brightness

Figure 1 shows the results of some corner detectors applied to the same image, shown in Fig. 1(a). This gray level image is 256 by 213 pixels coded on 8 bits. Figure 1(b) shows corners from the Moravec corner detector. Figure 1(c) shows corners from the Beaudet corner detector. Figure 1(d) shows corners from the Susan corner detector. Figure 1(e) shows corners extracted from the feature extraction (FEX) system (described later).

Representation Issues

The representation of a point feature, as extracted by the procedures already described, does not present specific difficulties. It should be noted, however, that while some methods identify pixels as point features, it is also possible to generate features with subpixel precision. One possible way to achieve this result is by fitting a continuous surface to the discrete intensity image. An instance of such an approach is presented by Zuniga and Haralick (8), using bicubic polynomials.

Also, it may be useful for certain applications to keep information about the feature in addition to its location. For instance, if the feature is identified as a vertex, one may use the type of the vertex (L, Y, Fork, T, . . .) and associated parameters (angles, contrast, color, . . .) as discriminating attributes for matching features.

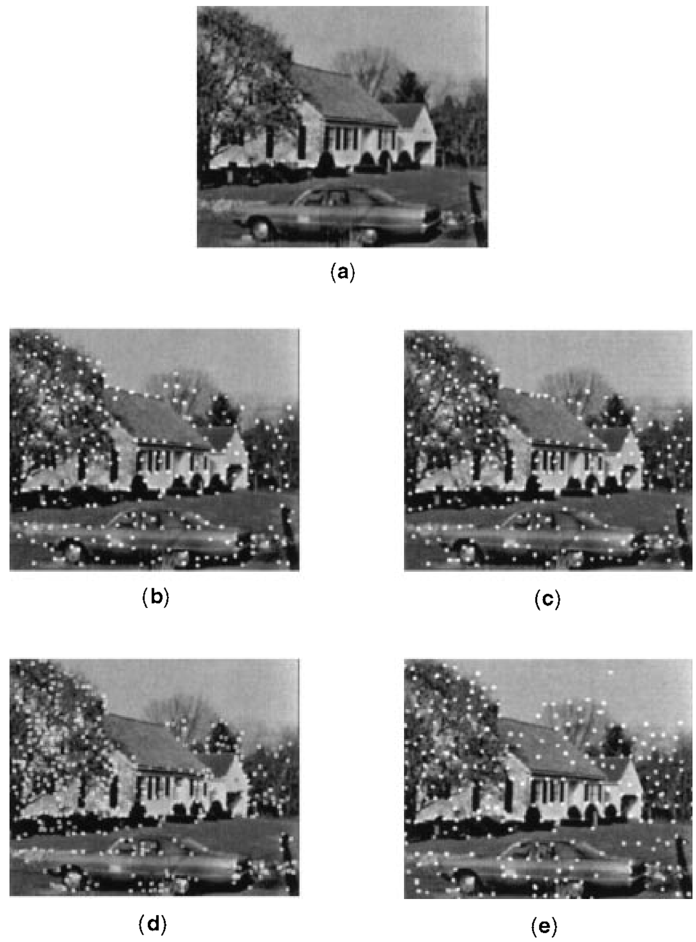


Figure 1. Corners from four different detectors. (a) Original image; (b) Moravec corners; (c) Beaudet corners; (d) Susan corners; (e) FEX corners. Corners correspond to extremal values of an expression computed in a small window. The results depend on the form of the expression and the size of the window.

EXTRACTION OF CURVE FEATURES

Detection of Edgels

From the very early days of computer vision, edge detection was recognized as an essential step, with a simple implementation of a gradient function (9). Noise sensitivity advocates the inclusion of a smoothing step before differentiation (10). This can be explained by the fact that differentiation is a typical ill-posed problem. The general theory of regularization (11) may be used to transform ill-posed problems into well-posed problems by restricting the class of possible solutions. Smoothing serves to regularize the input, making the differentiation operation mathematically well-posed.

The images can be either smoothed by convolution with some filters (Gaussian, for example), or approximated locally by a smooth analytic function. Therefore, we can roughly divide the edge detection techniques into two categories:

- Gradient estimation for edge detection is performed by first convolving the image with filters. For linear filters, the order of differentiation and convolution is interchangeable, so the image is convolved directly with the derivative of the smoothing filter. The filters can either be rotationally invariant, leading to closed zero-crossing contours, or directional, leading to better localization accuracy.

- Edge detection can be achieved by fitting a local surface expressed in terms of polynomials or splines, for example, in order to derive local properties (such as gradient) of the image, then to decide whether a point should be marked as an edge or not.

We start with the derivation of linear filters, and discuss in particular the Gaussian filter and its derivatives. We then discuss surface-fitting methods which use a variety of basis functions to perform the approximation. In these methods, a differentiation operation is then performed analytically on the approximation of the intensity function.

Linear Filters

The Gaussian Filter. By far the most popular smoothing filter is the Gaussian:

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

and its derivatives, the first derivative of the Gaussian:

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

the second derivative of the Gaussian:

$$g''(x) = \frac{1}{\sqrt{2\pi}\sigma^3}\left(\frac{x^2}{\sigma^2} - 1\right) e^{-\frac{x^2}{2\sigma^2}}$$

The popularity of the Gaussian filters as smoothing operators for edge detection comes from the following facts:

Optimality. Marr and Hildreth (12,13) argue that smoothing should have both limited support in the spatial domain and limited bandwidth in the frequency domain. In general terms, limited support in the spatial domain is important because the physical edges to be detected are spatially localized, and a limited bandwidth in the frequency domain provides a means of restricting the range of scales over which intensity changes are detected. The Gaussian function minimizes the product of the bandwidths in spatial and frequency domains. The smoothing functions that do not satisfy the limited bandwidths in space and frequency can sometimes lead to poorer performance, reflected in sensitivity to noise, detection of edges that do not exist, or poor ability to localize the position of edges. In two dimensions the basic approach is to convolve the signal with a rotationally symmetric Laplacian-of-Gaussian mask (sometimes approximated by a Difference-of-Gaussians), and to locate zero-crossings of the convolution. A paper by Torre and Poggio (14) judiciously points out that better results may be obtained by using two-directional filters with directional derivatives, especially in the neighborhood of corners.

Nice Scaling Behavior. Image events occur at different resolutions or scales, and Gaussian filters are the only filters having nice scaling properties, that is, no new events occur as scale increases.

Convolution with the Gaussian filters has the well-known property of constituting a solution to the following heat equation (15):

$$\frac{\partial}{\partial t} I(x, y, t) = c\nabla^2 I(x, y, t) \tag{7}$$

with initial condition $I(x, y, 0) = I_0(x, y)$, the original image. And this is the foundation of the Gaussian scale space which has been widely used in multiscale description of images (15–18). The essential idea is to embed the original image in a family of derived images $I(x, y, t)$ obtained by convolving the original image $I_0(x, y)$ with a Gaussian kernel $G(x, y, t)$ of variance t. The parameter t, bearing the physical meaning of time in the heat diffusion equation, serves as the scale parameter in the scale space paradigm.

Efficient Algorithm for Computation. Another reason for the popularity of Gaussian filters comes from practical considerations, that is, the convolution of an image with a Gaussian filter can be computed very efficiently, whether by approximation or not. There are other optimal filters which have been derived by vision researchers, but most are very similar to the Gaussian filters. The computational efficiency has made Gaussian filters very good approximations of their smoothing filters.

Figure 2 presents the results of convolving the image of Fig. 1(a) with Laplacian-of-Gaussian filters of different sizes. The edges are on the transition curve between the black and white regions, corresponding to positive and negative responses, respectively. We present results at the following scales: $\sigma = \sqrt{2}$, $1.5\sqrt{2}$, $2\sqrt{2}$ and $3\sqrt{2}$.

Optimal Frequency Domain Filter. Shanmugan, Dickey, and Green (19) obtain a frequency domain band-limited filter which concentrates maximal energy near an (ideal step) edge. Following Slepian, Pollak, and Landau (20,21), they decompose the optimal filter in terms of prolate spheroidal wave functions and show that the optimal filter output is ψ_1, the first order prolate spheroidal wave function, with space-bandwidth parameter depending on the space bandwidth cutoffs

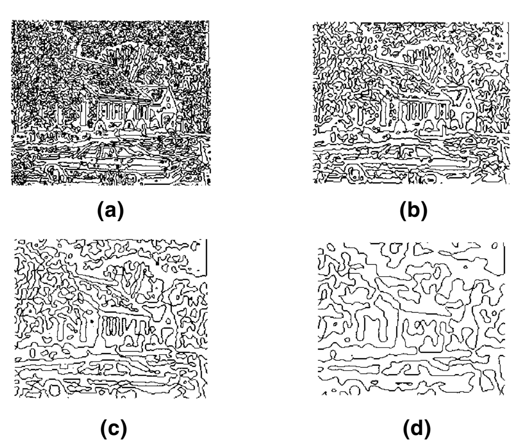


Figure 2. Edges from a Laplacian-of-Gaussian filter at different scales. (a) $\sigma = \sqrt{2}$; (b) $\sigma = 1.5\sqrt{2}$; (c) $\sigma = 2\sqrt{2}$; (d) $\sigma = 3\sqrt{2}$. As σ increases, more smoothing is applied, and fewer edges are detected; also, noise tolerance increases at the cost of a loss in localization.

required. This method, unlike the situation with a Gaussian, allows the space and the bandwidth cutoff to be chosen independently. Specifically, the transfer function of the optimal filter is given by:

$$H(\omega) = K \times \frac{\psi_1 \times \left(\Omega I, \dfrac{\omega I}{\Omega}\right)}{i \times F(\omega)} \qquad if\ \|\omega\| < \Omega$$

and $H(\omega) = 0$ if $|\omega| \geq \Omega$, where K is a real constant, ψ_1 is the first order prolate spheroidal wave function, Ω is the half bandwidth (i.e., the signal is nonzero only when $\omega \in (-\Omega, \Omega)$), the energy is to be concentrated in the spatial interval $(-I, I)$, and $F(\omega)$ is the Fourier transform of the ideal input. Using an approximation proposed by Slepian, the optimal filter within the passband can be approximated by:

$$H(\omega) = K_1 \times \frac{\omega \times e^{-K_2 \times \omega^2}}{i \times F(\omega)} \tag{8}$$

where K_1 and K_2 are simple functions of Ω and I. When the input is an ideal step edge, this reduces to:

$$H(\omega) = K_1 \times \omega^2 \times e^{-K_2 \times \omega^2} \tag{9}$$

This is equivalent to a second derivative of a Gaussian (i.e., Laplacian-of-Gaussian) with proper choice of K_1 and K_2 to match the variance σ of the Gaussian.

The Canny Edge Detector. Canny (22) proposes an edge detection scheme based on efficiency of detection and reliability in localization. He seeks an optimal filter satisfying the following criteria:

Good Detection. The edge detector should have a low probability of failing to detect real edge points, and also low probability of falsely marking nonedge points as edges. Both these probabilities are monotonically decreasing functions of the output signal-to-noise ratio, the good detection criterion corresponding to maximizing the output signal-to-noise ratio.

Good Localization. The points that are marked as edge points by the edge detector should be as close as possible to the actual location of the edge points.

Only One Response to a Single Edge. This criterion is somewhat implicitly captured in the good detection criterion since when there is more than one response to the same edge, only one can be considered as true edge and the other responses must be considered false. However, the mathematical form of the first criterion does not capture the multiple response requirement and it has to be made explicit.

Canny poses the 1-D edge detection problem as an optimization problem over the set of convolution operators, which he solves by the use of both variational and numerical methods. He derives the optimal operator, in one dimension, which is a linear combination of four exponentials:

$$\begin{aligned} f(x) = {} & a_1 e^{\alpha x} \sin(\omega x) + a_2 e^{\alpha x} \cos(\omega x) + a_3 e^{-\alpha x} \sin(\omega x) \\ & + a_4 e^{-\alpha x} \cos(\omega x) + c \end{aligned} \tag{10}$$

and is subject to the boundary conditions:

$$f(0) = 0,\ f(-W) = 0,\ f'(0) = s,\ f'(-W) = 0$$

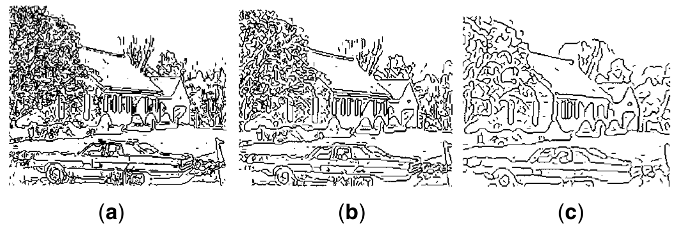


Figure 3. Edgels from the Canny operators at three different scales. (a) Filter size = 4; (b) Filter size = 8; (c) Filter size = 16. Different size masks extract edges at different scales.

where s is an unknown constant equal to the slope of the function f at the origin. The Canny operator can be well approximated by a first derivative of a Gaussian, which can be computed efficiently not only in 1-D, but also when extended to 2-D. Canny does not directly consider a 2-D optimization problem for deriving the optimal filter in 2-D; instead he starts from the point he reached with the 1-D problem. The approach is to use an operator of the form $h(x, y) = f(x) \times g(y)$, for various orientations of the orthogonal coordinates, where g is a Gaussian. Deriche (23) develops a recursive filter for edge detection using Canny's criteria. The filter can be implemented efficiently because of its recursive nature and therefore it does not need the Gaussian approximation, such as used by Canny, for fast computation.

We display the results of the Canny edge detector for the picture in Fig. 1(a) on Fig. 3, using different scales. These scales correspond to the following values of filter size: 4, 8, and 16.

Surface Fitting. Surface modeling of image data defines another category of edge detectors. This method involves an initial parameterization of the local image data in terms of some set of basis functions, followed by the estimation of the contrast and location of the best-fit step edge from the parameters. The idea is to first find a continuous function whose samples correspond to the discrete image. It is then mathematically appropriate to compute derivatives and other differential properties which, strictly speaking, cannot be obtained from discrete data.

The Prewitt Operator. Prewitt (24) discussed a wide range of image understanding problems from image formation to object extraction. She used a quadratic set of basis functions for surface fitting. The so-called Prewitt operator is a 3×3 window to estimate the gradient, which give $\partial/\partial x$ and $\partial/\partial y$ for that surface directly from the data. She also discusses oriented edge masks as approximations to the gradient. The Prewitt operator for estimating the gradient is based on the best fit of a plane. However, since the support is very small, global structures cannot contribute to the edge finding process, and the derived image description is limited to one or two local parameters, which is in general not sufficient.

The Hueckel Operator. Hueckel (25,26) applied basis functions with circular support and tried to fit a single-step edge for each circular area. The basis functions are chosen so as to give an approximate Fourier transform of the circular region which is 32 to 137 pixels. In edge fitting, the image function $I(x, y)$ defined over a support D is compared with an ideal edge model $M_p(x, y)$, where p is the edge parameter vector. The error difference is given in the form

$$E_p = \sum\sum(I(x,y) - M_p(x,y))^2 \quad (11)$$

Hueckel used an orthogonal transform to solve the edge fitting problem. In particular, the error is given by

$$E_p = \sum_{i=0}^{N}(a_i - s_i)^2$$

where

$$a_i = \sum\sum H_i(x,y)I(x,y)$$

and

$$s_i = \sum\sum H_i(x,y)M_p(x,y)$$

Theoretically, N should approach infinity, but the approximation can be made using a truncated form. The orthonormal expansion H_i's used consists of polynomials in x, y with a uniform radial weighting function $\sqrt{1 - x^2 - y^2}$. For the edge operator, eight polynomials of degree up to three are used (25), while the edge-line uses nine polynomials of degree up to four (26). The edge parameter vector $p_{\min}$ that minimizes the truncated form ($N = 7$ or $N = 8$) can be found by solving simple algebraic equations. The edge/no-edge decision is based on the angle between the projections of the data and the best-fit edge in the truncated space. Abdou (27) presented a detailed analysis of Hueckel's operator, and noticed that the difficulties with the operator came from the truncation of the series expansion, inaccuracies in the minimization procedure, and the computation of the edge parameters.

The Haralick Operator. In his 1980 article, Haralick (28) proposes to fit the image data by small planar surfaces or facets. Edges are marked at points which belong to two such facets when the parameters of the two surfaces are inconsistent. The test of consistency is based on the goodness of fit of each surface within its neighborhood and uses a χ^2 statistic. The statistics become more complicated for more complicated fits. A more general surface fitting technique is used in his later work (29). He used higher order polynomial basis functions with larger operator supports. He imposes 1-D symmetry on the index sets of the polynomials, that is, the points at which they are defined must be symmetric about the origin. He uses the tensor product of his 1-D set to define the 2-D basis functions. He then shows how to fit by the method of projection onto the orthonormal basis. His definition of edges are the zero-crossings of the second directional derivative in the gradient direction, namely the maximum of the gradient. The choice of polynomials for the facet model is basically on the ground that they are easier to manipulate. The degree of approximation of polynomials is poor especially at discontinuities or edges in the images. In their 1985 article, Watson, Laffey, and Haralick (30) propose a general spline approach to improve the performance of the facet model.

The Nalwa Operator. Instead of marking pixels as belonging to an edge, Nalwa (31) defines an edge in terms of edgels, that is, short linear segments, each characterized by an orientation and a position, and corresponding to discontinuities in the image data. He fits to the window a series of 1-D surfaces, that is, surfaces constrained to be constant in one dimension, and accepts the surfaces with the smallest adequate basis. He uses *tanh* as an adequate basis for a step edge, and its combinations are adequate for the roof and the line edges. He also takes into account the blurring function of the imaging system, which is a Gaussian function, to a first-order approximation. It is a very complex algorithm as far as edge detection is concerned. It first fits a planar surface to the window and minimizes the square-error, followed by a 1-D cubic surface fit with the same error criteria to refine the estimate. It then fits an optimal tanh 1-D surface compared to a quadratic fit and uses F-statistic to determine the existence of the edgel. The process is repeated for each pixel location in the image. He claims that his approach is robust with respect to noise, and for (*step-size* $1/\sigma_{\text{noise}}$) ≥ 2, it has subpixel position resolution and a 5° angle resolution.

Contour Representation

Most edge detection methods, as presented in the last section, produce edgels which need to be further aggregated into chains. This grouping step is called linking, and has received much less attention that the detection part. Nevatia (32) proposes a local heuristic method seeking the most compatible candidate in a small neighborhood, and handles junctions. Zero-crossing edge detectors are easier to handle, as they are guaranteed to produce closed contours and no junctions. The resulting chain can be extracted with subpixel accuracy (33).

Once a chain of points has been extracted, we need to address the issues of representation. A digital curve can simply be represented by a linked list of its component pixels. While complete, this description is cumbersome and wasteful. Another complete representation is the chain code: given two adjacent pixels (x_i, y_i) and $(i + 1, j + 1)$ of the curve, it is sufficient to represent the direction changes between the ith and $i + 1$ pixel, as there exist only eight possible such direction changes. Many algorithms have been designed to operate directly on such a representation (34).

Rather than describing the curve in terms of pixels, it may be useful to approximate the curve in terms of higher level primitives, such as linear segments, or curved segments, typically low order polynomials. The issue to be addressed relates to the selection of breakpoints between these primitives, a step also called corner detection.

A simple approach (35) consists of using a single line segment to approximate a curve, then of recursively splitting the curve into two subcurves at the point maximally distant from the line. It is also possible to use a merging algorithm, which iteratively builds longer segments until an error threshold is exceeded. Another approach consists of mapping edgels onto a representation which consists of the arc length and the tangent (36), as straight lines map to horizontal straight lines. The estimation of the tangent value, however, is difficult. For a detailed discussion and implementations, see the book by Pavlidis (37). In principle, these methods can be extended to fit with higher level primitives, such as conic sections, but the estimation of distance to the curve is difficult, and the fit may be unstable or biased.

The methods already discussed produce corners as a result of the fitting procedure. A number of methods instead propose to first detect points of maximum curvature, then to perform a fit between consecutive ones. Curvature estimation is a numerically delicate operation, which can be performed using the edgel chain (38–40) or directly from the partial derivatives of the image with respect to x and y (41). We present some details of a method to fit a curve to a set of data points using B-splines (42).

Given C, an ordered set of $p + 1$ points $P_i = (x_i, y_i)$, we look for the B-spline which best approximates C. The approach proposed in Ref. 43 consists of minimizing the distance

$$\begin{aligned} R &= \sum_{i=0}^{p-1} \|Q(u_i) - P_i\|^2 \\ &= \sum_{i=0}^{p-1} \left[\left(\sum_{j=0}^{m-1} X_j B_j(u_i) - x_i \right)^2 + \left(\sum_{j=0}^{m-1} Y_j B_j(u_i) - y_i \right)^2 \right] \end{aligned} \tag{12}$$

and u_i is some parameter value associated with the ith data point. Minimizing R is equivalent to setting all partial derivatives $\partial R/\partial X_l$ and $\partial R/\partial Y_l$ to 0, for $0 \le l < m$, which yields

$$\begin{aligned} \sum_{j=0}^{m-1} X_j \sum_{i=0}^{p-1} B_j(u_i) B_l(u_i) &= \sum_{i=0}^{p-1} x_i B_l(u_i) \\ \sum_{j=0}^{m-1} Y_j \sum_{i=0}^{p-1} B_j(u_i) B_l(u_i) &= \sum_{i=0}^{p-1} y_i B_l(u_i) \end{aligned} \tag{13}$$

with $0 \le l < m$.

These linear systems are easily solved for all X_j and Y_j using standard linear algebra, yielding the guiding polygon of the B-spline which best approximates the original curve. The choice of m (the number of vertices) determines how close to the original data the approximation is, which is measured by R.

Adapting a Predefined Pattern to the Data

In many applications, we are looking at an image for a specific pattern, which can be defined by a curve or a set of curves. Depending on whether we have an initial guess of the position of the curve pattern or not, we can use different methods. We first study the problem of estimating the parameters of a known curve pattern to a set of data points, in the presence of noise and outliers. We then show how to extract a smooth curve, whose exact equation is not known, given an initial estimate of its position.

The Hough Transform. The Hough transform (44,45) is a method which allows detection of instances of a pattern whose analytic expression is known, by working in the space of the parameters instead of the image space. The method is most useful for the detection of shapes defined by three or less parameters, such as straight lines or circles. The method uses an accumulator array of dimensions equal to the number of parameters of the family of shapes being sought. For instance, straight lines require two parameters, and the dimension of the array is two. Circles require three parameters (coordinates of the center, and radius), so the accumulator dimension is three. We next describe in more details the procedure to detect lines given a set of points.

The general equation of a straight line can be written in polar form as

$$x \cos\theta + y \sin\theta = \rho$$

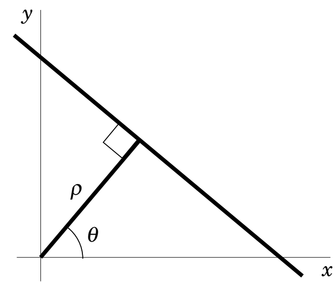


Figure 4. Polar representation of a line. This representation avoids problems created by infinite values in slope-intercept form.

where θ is the angle between the line and the x-axis, and ρ is the distance from the origin to the line. Figure 4 illustrates this relation between x, y, ρ, and θ.

Any line passing through the point (x_i, y_i) must satisfy

$$x_i \cos\theta + y_i \sin\theta = \rho$$

This equation defines a sinusoidal curve in (ρ, θ) space, corresponding to the Hough transform of point (x_i, y_i) into the (ρ, θ) space. Collinear points in Cartesian space produce curves which intersect at a single point in (ρ, θ) space. Figure 5 shows the two curves in (ρ, θ) space corresponding to the two points $(-1, -1)$ and $(1, 1)$, which intersect at $\rho = 0$ and $\theta = 135°$.

This observation is the key to the following implementation:

- Quantize the (ρ, θ) space
- For each point (x_i, y_i), generate the corresponding digitized curve in (ρ, θ) space
- For each point (ρ_i, θ_i) of this curve, increment the counter at this location
- Locations in (ρ, θ) with high counter values correspond to the desired lines

The critical issues in the implementation of the Hough transform relate to the choice of the quantization parameters: a coarse quantization produces poor localization, and a fine quantization leads to poor noise tolerance.

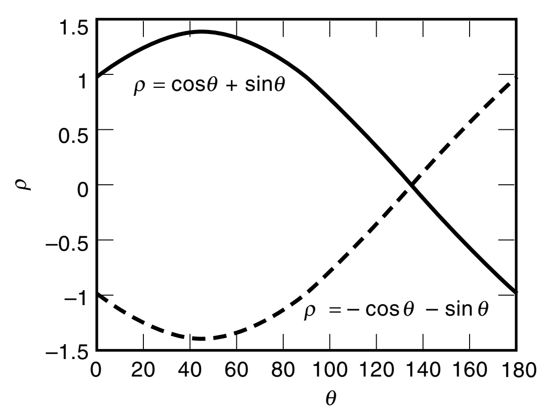
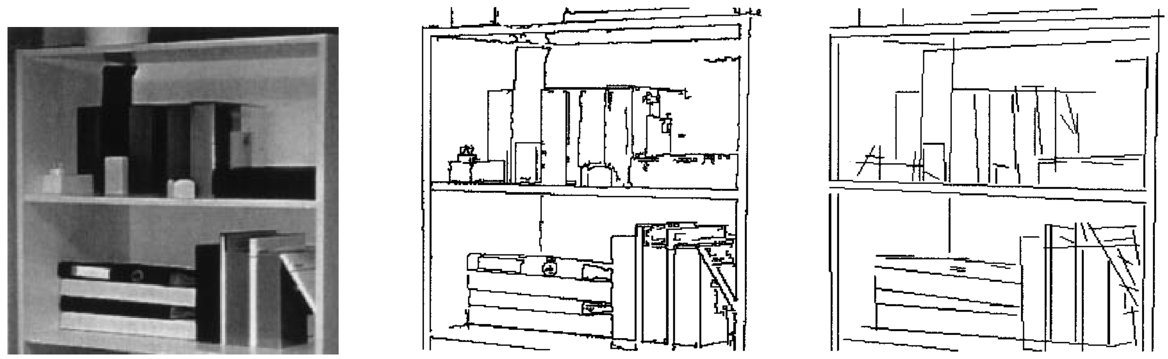


Figure 5. Hough accumulator from two points, (1, 1) and (1, −1). Each point produces a sinusoidal curve in the accumulator space. The intersection between these curves corresponds to the line joining the two points.



(a) Original Image *(b) Edgel Image* *(c) After Hough Transform*

Figure 6. Example of line detection using the Hough transform. (a) Original image; (b) Edgel image; (c) After Hough transform. A filtering step is needed in (c) to clip the lines produced by the Hough transform.

Many improvements have been suggested to improve the basic technique, including probabilistic formulation (46) and a randomized approach (47). We refer the reader to Ref. 48 for a recent overview.

We present the results of applying the Hough transform to detect straight lines in Fig. 6. Figure 6(a) shows the original image. Figure 6(b) shows the edges extracted. Figure 6(c) shows straight lines extracted from the Hough transform.

Snakes. Active contour models were introduced in Ref. 49 as a methodology to deform a predefined curve under a set of forces. The forces result from an internal energy describing the elasticity of the curve, and from an external energy describing the quality of the fit. This active contour model fits in an interactive human–machine environment when the user supplies an initial estimate of the object to extract, and the snake is used to refine the results (49,50). It is also useful when a first estimate is given by a prior processing level. We will next describe the equations needed to implement such a scheme.

A snake is a deformable, continuous curve, whose shape is controlled by internal forces (the implicit model) and external forces (the data). Internal forces act as a smoothness constraint, and external forces guide the active contour towards image features.

Let $v(s) = (x(s), y(s))$ be the parametric description of the snake ($s \in [0, 1]$). Its total energy can be written as:

$$\begin{aligned} E_{\text{snake}} &= \int_0^1 E_s(v(s))\,ds \\ &= \int_0^1 [E_{\text{int}}(v(s)) + E_{\text{ext}}(v(s))]\,ds \end{aligned} \tag{14}$$

with:

$$E_{\text{int}}(s) = \frac{1}{2}(\alpha(s)|v_s(s)|^2 + \beta(s)|v_{ss}(s)|^2)$$

We seek the snake that minimizes the energy E_{snake}, given some external energy adapted to image features to extract ($E_{\text{edge}} = -|\nabla I(x, y)|^2$, for example) and internal energy whose expression was previously given. The first order term makes the snake act like a membrane and the second order one like a thin plate. This energy is the regularizing term of the minimization.

The minimization of E_{snake} is solved by using the calculus of variations and resolving Euler equations, and yields the following equations in the discrete case (49):

$$\begin{cases} A_x + F_x(x, y) = 0 \\ A_y + F_y(x, y) = 0 \end{cases} \tag{15}$$

where $F = E_{\text{ext}}$ depends on the image features to extract and A is a pentadiagonal matrix depending on α and β.

This system of equations in (x, y) is solved by introducing an energy dissipation functional to dissipate the kinetic energy during the motion. Let γ be the Euler step size. The expression of the snake as a function of time is then:

$$\begin{cases} x_{t+1} = (A + \gamma I)^{-1}(\gamma x_t - F_x(x_t, y_t)) \\ y_{t+1} = (A + \gamma I)^{-1}(\gamma y_t - F_y(x_t, y_t)) \end{cases}$$

$(A + \gamma I)^{-1}$ can be calculated by LU decomposition (a product of a lower and upper triangular matrices) in $O(n)$ time (n is the length of the snake).

The convergence rate of a snake using all points can be slow, so authors have proposed to represent the curve by a B-spline instead of points (51). The equations are the same, but the number of points is reduced, leading to stability.

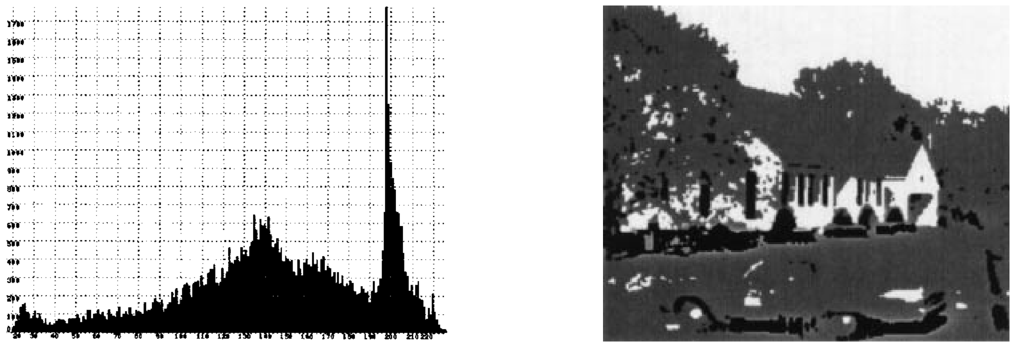
EXTRACTION OF REGION FEATURES

Unlike the preceding methods, which aim at finding boundaries between regions sharing one or more common properties, region segmentation procedures aggregate adjacent pixels into connected components by splitting, merging, or a combination of these two operations.

We briefly describe these procedures next.

Detection of Regions

Thresholding. The simplest possible method to generate a set of regions is by means of thresholding. This consists of



(a) Histogram *(b) After thresholding*

Figure 7. Region segmentation using two fixed thresholds. (a) Histogram; (b) After thresholding. The two threshold values, 60 and 190, were chosen by hand, and correspond to valleys in the histogram.

partitioning the set of gray levels into a coarser set of intervals, and of assigning each pixel with a given value to the corresponding class. Thresholding methods are appropriate for scenes in which high-contrast objects are imaged in front of a uniform background, for instance, characters on a page. Clearly, the central issue to be addressed is the choice of the threshold values. These may be manually estimated from a training set of images, or from known operational conditions. Instead, these thresholds can be derived by considering the histogram of the image. Homogeneous regions should produce peaks in the histograms, so potential thresholds should be chosen in valleys between peaks. Several problems are associated with histogram-based methods: first, the extraction of valleys may be difficult, as there may be many such minima, and some of these may be flat. Second, pixels in the same class may not form a coherent region, but a large number of small ones, as the spatial distribution of gray-levels is ignored. Third, boundaries of the regions may not coincide with contours, again because spatial information is not used. A number of methods have been proposed to reduce, but not eliminate, these problems. Of particular interest is the *superspike* technique (52) which performs iterative local averaging to produce sharper histogram peaks. Also of interest is the approach of Pappas and Jayant (53) which uses multiple size windows, and imposes a Markov random field model to enforce spatial coherence.

Figure 7 illustrates region segmentation by thresholding on the original image from Figure 1(a). Figure 7(a) shows the histogram of the original image. Figure 7(b) shows the segmented regions after using 60 and 190 as two thresholding values.

Recursive Segmentation. Rather than setting thresholds statically, it is possible to apply a recursive procedure: given an initial set of thresholds based on histograms, each resulting region is considered a new image, for which new thresholds can be derived. The process is repeated until no new peaks can be isolated, or the regions become too small. An excellent implementation is described in Ref. 54. If the input is a color image, then the method computes histograms in different spaces, and selects at each iteration the most prominent peak.

Region Growing. Region growing (or merging) methods, proceed from a set of seed regions, either individual pixels, or groups of pixels with nearly identical properties, and absorb neighbors of a region by comparing their relative properties. These may include, besides average intensity, criteria such as regularity of shape. Regions grown from seed regions may overlap, and therefore produce a description which is not a partition of the image. Examples of such schemes can be found in Ref. 55. Also worth noting is the approach of Besl and Jain (56) which decides whether a set of pixels belongs to a region by comparing least-squares errors of fitting multiple-order bivariate polynomial surfaces, up to the fourth order, to the pixel values.

Split and Merge Segmentation. It is possible to combine the ideas already expressed into a split and merge scheme, as described in Ref. 57, in which, given an arbitrary initial partition in a pyramidal data structure, adjacent regions having similar approximations are merged, and regions with a high-error norm are split. To achieve the partition, we first create a pyramid representation of the image, in which a pixel at a given level represents four pixels at the level below. At the top, the whole image is encoded as a single region, and at the bottom, each region is a pixel of the original image. The algorithm then proceeds by moving up and down in this pyramid, performing merging and splitting operations, which decrease the approximation error norm, until a minimum is attained.

Region Representation

Given a connected set of pixels forming a digital region, the issue of representation arises again. We can represent a region by its bounding curve(s), in which case the representation issues are identical to the ones covered in the section titled "Contour Representation."

A region can be represented by a list of points belonging to it, or by a spatial occupancy array, which is a binary mask taking values 1 for pixels inside a region, and 0 otherwise. A very useful encoding of such an array can be performed using quadtrees (58): given an image, one builds a pyramid by successively reducing resolution by a factor two. A pixel at any level above the original image level can be assigned one of three values, Black, White, or Gray. A pixel is Black (respectively White) if all four corresponding pixels at the lower level are Black (respectively White), Gray otherwise. Many operations, such as union, intersection, and genus, can be efficiently computed using quadtrees, as complexity depends on the number of blocks rather than on the number of pixels. The main drawbacks are that quadtrees are not invariant under translation or scale shifts.

A region can also be represented by a covering consisting of the set of maximal disks which touch at least two points of the boundary (59,60). Given the loci of the disk centers and their radii, the shape can be regenerated. Such a representation is called the symmetric axis transform (SAT), medial axis transform, or Blum transform. While it is an elegant representation with interesting properties (uniqueness, generative capability), it has rarely been used in applications, probably because of its instability: a small change in the boundary may create major changes in the axes.

Finally, it is worth noting that, for some applications, approximations of the shape may be sufficient. Properties such as perimeter, area, compactness (perimeter^2/area), elongation (ratio of maximum chord A to maximum chord B perpendicular to A), Euler number, moments, and Fourier descriptors, have been used. Also popular are best fit by a parametric family of shapes, such as minimum bounding rectangle, bounding ellipse, and best-fit ellipse. A generalization of the quadrics leading to an interesting family of shapes called superquadrics was proposed by Hein and promoted by Barr (61). A superellipse is given by the equation $(x/a)^n + (y/b)^n = 1$, with $a, b, n > 0$.

COUPLED EXTRACTION OF POINTS, CURVES, AND REGIONS

Whereas the previous methods explicitly attempt to extract points, curves, or regions independently, it is possible to state the problem in a coupled form, as finding the optimal partition of the image into homogeneous regions, and its edges, or boundaries. What constitutes optimality crucially conditions the results. Mumford and Shah (62) define a so-called universal segmentation model, as a joint smoothing/edge detection problem: given an image $g(x)$, find a piecewise smoothed image $u(x)$ with a set K of discontinuities. The solution is obtained by minimizing

$$E(u,K) = \int_{\Omega \backslash K} (|\nabla u(x)|^2 + (u-g)^2)\,dx + \text{length}(K) \tag{16}$$

The first term imposes that u is smooth outside the edges, the second that the piecewise smooth image $u(x)$ indeed approximates $g(x)$, and the third that the discontinuity set K has minimal length (and therefore in particular is as smooth as possible).

An excellent analysis of the mathematical issues is presented in Ref. 63, showing that minimal segmentations exist, but are not unique. Furthermore, it claims that most segmentation methods can be interpreted as attempts to minimize this variational energy functional.

We next present four very different approaches to the coupled feature extraction problem: anisotropic diffusion (64) proposes to solve the heat equation discussed in the section titled "The Gaussian Filter" with discontinuities. Geman and Geman (65) use a stochastic formulation of the problem and impose a Markov random field (MRF) model on the image. Blake and Zisserman (66) perform surface reconstruction with discontinuities using an elegant algorithm. Finally, Förstner (67), in a system called FEX, produces an integrated description in terms of corners, curves, and regions.

Anisotropic Diffusion

Perona and Malik (64) point out that Koenderink (16) motivates the diffusion equation formulation by the following two criteria:

Causality. Any feature at a coarser resolution must be caused by an event at finer resolution, therefore no new features will arise when traversing the scale space from fine to coarse.

Homogeneity and Isotropy. The blurring is required to be space-invariant.

These criteria lead naturally to the formulation of the heat-diffusion equation. Gaussian smoothing has long been criticized because it not only reduces the noise but also smooths the edges and introduces inaccuracy to the edge localization. The Gaussian scale space therefore always has to deal with the correspondence problem in scale space. Perona and Malik point out the important observation by Hummel (15) that the maximal principle from the theory of parabolic differential equations is equivalent to causality in scale space. The causality criterion does not force the unique choice of the Gaussian kernel for smoothing. Perona and Malik (64) derive an algorithm by varying the conduction property of the material and still preserve the causality principle while performing the edge-preserving smoothing. In the standard heat equation already discussed, the material is considered to be homogeneous and the conduction property of the material is described by the constant c on the right-hand side of the heat equation. The idea of anisotropic diffusion is therefore to introduce inhomogeneity into the material and the smoothing process becomes a controlled diffusion process. The purpose is, of course, to prevent the diffusion or smoothing across edges. The anisotropic diffusion is formulated as follows:

$$\frac{\partial u(x,y,t)}{\partial t} = \nabla(c(x,y,t)\nabla u(x,y,t)) \tag{17}$$

Note that, if $c(x, y, t)$ is a constant, then Eq. (17) reduces to the isotropic heat-diffusion equation. The function $c(x, y, t)$ is intuitively chosen as a function of the gradient of the image at that point. Since the gradient at the region boundary tends to have higher gradient value, $c(x, y, t)$ should be small to prevent the diffusion across the boundary. On the other hand, if the gradient is small, the point is most likely to be inside a region and the diffusion is encouraged, therefore $c(x, y, t)$ is large at that point. Anisotropic diffusion differs from a standard iterative smoothing formulation in which each center pixel is replaced by a window of weighted neighborhood. It concerns the diffusion between neighboring pixels, that is, each pixel is updated by a certain amount depending on the gray levels between the center pixel and its direct neighboring pixels. Anisotropic diffusion provides a good tool for discontinuity preserving smoothing and it is well formulated mathematically. The well-developed mathematical tools in the area of partial differential equations, especially the

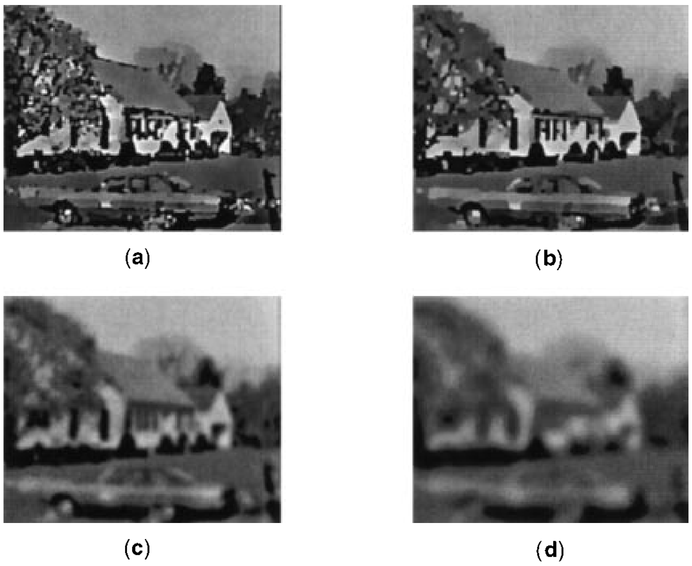


Figure 8. Gaussian and adaptive smoothing at 2 different scales. (a) After adaptive smoothing ($k = 12$); (b) After adaptive smoothing ($k = 24$); (c) After Gaussian smoothing ($\sigma = 2$); (d) After Gaussian smoothing ($\sigma = 4$). Gaussian smoothing is uniform, whereas adaptive smoothing sharpens boundaries while smoothing regions.

heat-diffusion equation, make anisotropic diffusion easy to analyze. In anisotropic diffusion, however, we assume that edges are perfect steps and therefore cannot directly deal with other type of edges such as roof edges. Also, anisotropic diffusion needs a very large number of iterations to reach its final convergent state, but this problem can be handled by considering that edges do not change after a few iterations.

Figure 8 shows the results of Gaussian and adaptive smoothing at two different scales. Figure 8(a) and (b) show the results after adaptive smoothing with k set to 12 and 24 respectively. Figure 8(c) and (d) show the result of Gaussian smoothing with σ set to 2 and 4 respectively.

Stochastic Approach

Geman and Geman (65) link together mechanical systems like soap films, splines, and statistical theory. They use the MRF model as the formalism for describing images, establish theorems that provide a means for specifying the probability of a particular original image given the observed degraded data, and use the technique of simulated annealing as a mechanism for finding the image that maximizes the probability of it being the replica of the original image given the observed data. Finally, a model of spatial coherence in images is introduced to allow the placement of image boundaries that terminate this coherence. A Markov random field is a lattice of pixels and each pixel can be assigned any of its allowed values. The conditional probability for a pixel having a certain value, given the values of all the other pixels in the image, is only a function of the pixels in a finite neighborhood of that pixel. These conditional probabilities specify uniquely the probability that the system is in a particular state. The probability is a Gibbs distribution whose form is particularly simple to compute. In particular,

$$Probability = \frac{1}{N} e^{-\frac{U}{T}} \tag{18}$$

where N is a normalization number, T is a (temperature) parameter, and U is the sum of potentials that specify how each neighbor, pair of neighbors, and so on contribute to the probability that the pixel has a certain value. The potential function U usually consists of two terms: one associated with the interaction potential and one associated with the difference between the predicted image and the observed data. Geman and Geman assume, in their image model, the expected similarity between gray-level intensity values of the neighboring pixels. They also assume edges may occur between pixels and that edge contours are assumed to be lines. The image restoration problem is therefore to find a set of pixel values and a set of line values that maximize the a posteriori probability of that state given observed data.

Weak Continuity Constraint

Blake and Zisserman (66) introduce the concept of weak continuity constraints to allow discontinuities in surface reconstruction. Following Blake and Zisserman's notation, the behavior of an elastic string over an interval $[0, N]$ is defined by an energy, which is a sum of three components:

1. The Penalty P measures the sum of penalties α levied for each break (discontinuity) in the string.
2. The Difference D measures the faithfulness to data.
3. The Smoothness S measures how severely the function $u(x)$ is deformed.

The problem is to minimize the total energy:

$$E = D + S + P$$

The finite element method allows converting the continuous problem to the following discrete problem:

$$E = \sum_{i=1}^{n}(u_i - d_i)^2 + \lambda^2 \sum_{i=1}^{n}(u_i - u_{i-1})^2(1 - l_i) + \alpha \sum_{i=1}^{n} l_i \tag{19}$$

where l_i is a so-called line-process. It is defined such that each l_i is a Boolean-valued variable: either $l_i = 1$, indicating that there is a discontinuity in the interval $[i - 1, i]$ or $l_i = 0$ indicating continuity in that interval. The solution of this approximation problem can be found by the following iterative scheme of successive over-relaxation:

$$u_i^{(t+1)} = u_i^{(t)} - \frac{\omega}{T}\frac{\partial E}{\partial u_i} \tag{20}$$

where $0 < \omega < 2$ is the successive over-relaxation (SOR) parameter, governing the speed of convergence, and T_i is an upper bound on the second derivative:

$$T_i \geq \frac{\partial^2}{\partial u_i^2} E$$

Since the energy function to be minimized is in general not convex, the system u_i may have many stable states, each corresponding to a local minimum of the energy function. The stable state reached usually depends on the initial state of the system. It is unlikely for that stable state, that is, the local minimum, to be the global minimum unless the convexity of the energy function is guaranteed. Blake and Zisserman come up with an elegant model called graduated nonconvexity (GNC) for minimizing the above weak continuity constraint equation. The basic idea is to find a sequence of approximating functions $E^{(p)}$, where p varies continuously from 1 to 0 (for practical purposes, p is set to be 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, . . .), $E^{(1)}$ is a convex function and $E^{(0)}$ is the original energy function to be minimized. Starting from $E^{(1)}$, its convexity guarantees that the local minimum is also the global minimum. The iteration proceeds as follows: the minimum found for $E^{(1/m)}$ is used as the initial value to find the minimum of $E^{(1/2m)}$. The basic model of weak continuity constraint measures the smoothness constraint with the first derivatives of the reconstruction, they are called *weak string* and *weak membrane* for 1-D and 2-D respectively. Blake and Zisserman also consider second derivatives to the energy term in smoothness constraint for *weak rod* (1-D) and *weak plate* (2-D). The visual reconstruction with weak continuity constraint model can be applied to a wide range of image data. The results are very impressive, although a large number of iterations is required to minimize the energy function. Furthermore, the causality criterion is not respected and new events can therefore occur at higher scales.

FEX

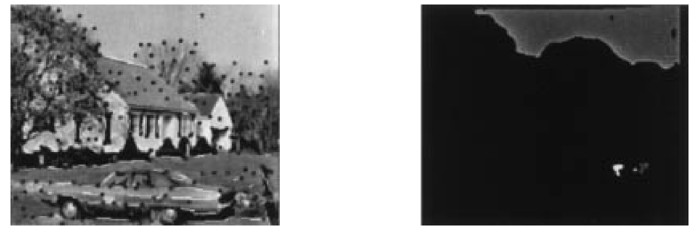
Förstner (67) presents a framework for extracting low-level features, namely points, edges, and segments from digital images. It is based on generic models for the scene, the sensing, and the image. Feature extraction is based on local statistics of the image function, and on the regularity of the image function with respect to junctions. Förstner provides methods for blind estimation of a signal dependent noise variance, for feature preserving restoration, feature detection and classification, and for the precise location of general edges and points. In all steps, thresholding and classification are based on proper test statistics, reducing threshold selection to choosing a significance level.

More specifically, he computes at each pixel the average squared gradient. He computes at every point the average square gradient matrix $\overline{\Gamma}\sigma\, g(x, y)$ as the convolution of a rotationally symmetric Gaussian and the square gradient matrix Γg. Γg is defined as the dyadic product

$$\nabla g \nabla g^T = \begin{pmatrix} g_x^2 & g_x g_y \\ g_y g_x & g_y^2 \end{pmatrix}$$

and $\nabla g = (g_x, g_y)^T$ is the image gradient. He interprets the trace of this matrix as a measure of homogeneity, the ratio of the eigenvalues as a measure of isotropy, and the largest eigenvalue as a measure of the local gradient. He then defines a regularity measure by analyzing the local gradient field, and estimates noise characteristics. Such an analysis leads to the classification of all image pixels into three classes—points, lines, and regions—which are then refined to optimize the regularity measures. He shows that the system behaves well at different scales.

Figure 9 shows corners, edges, and blobs extracted from the FEX system. Figure 9(a) shows both corners and edges extracted from the FEX system. Figure 9(b) shows blobs extracted from the FEX system.



(a) (b)

Figure 9. Results of the FEX system. (a) Corners + edges; (b) Blobs. All three types of features are extracted simultaneously.

CONCLUSION

Feature extraction is a central issue in the design and implementation of any vision system. As a result, the literature on the subject is varied and abundant. We have provided a broad overview of the issues involved in selecting proper features for a given application, and given details on some of the more popular strategies.

BIBLIOGRAPHY

1. H. Moravec, Towards automatic visual obstacle avoidance, *Int. J. Conf. Artif. Intell.,* Cambridge, MA, 1977, p. 584.
2. P. R. Beaudet, Rotationally invariant image operators, *Int. Conf. Pattern Recognition,* Kyoto, pp. 579–583, 1978.
3. L. Kitchen and A. Rosenfeld, Gray level corner detection, *Pattern Recognition Lett.,* **1** (2): 95–102, 1982.
4. J. A. Noble, Finding corners (image edge detection), *Image Vision Comput.,* **6** (2): 121–128, 1988.
5. C. Harris and M. J. Stephens, A combined corner and edge detector, *Alvey Conf.,* Manchester University, 1988, pp. 147–152.
6. R. Deriche and G. Giraudon, A computational approach for corner and vertex detection, *Int. J. Comput. Vision,* **10** (2): 101–124, 1993.
7. S. Smith and M. Brady, SUSAN—A new method for low level image processing, *Int. J. Comput. Vision,* **23** (1): 45–78, 1997.
8. O. A. Zuniga and R. M. Haralick, Corner detection using the facet model, *Proc. IEEE Comput. Vision Pattern Recognition,* 1983, pp. 30–37.
9. L. G. Roberts, Machine perception of three dimensional solids, *Optical and Electro-Optical Information Processing,* Cambridge, MA: MIT Press, 1965, pp. 562–569.
10. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene analysis, *IEEE Trans. Comput.,* **20**: 562–569, 1971.
11. A. N. Tikhonov and V. Y. Arsenin, *Solutions to Ill-Posed Problems,* Washington, DC: Winston, 1977.
12. D. Marr and E. C. Hildreth, Theory of edge detection, *Proc. R. Soc. London,* **B207**: 187–217, 1980.
13. E. C. Hildreth, The detection of intensity changes by computer and biological vision systems, *J. Comput. Vision, Graphics Image Process.,* **22**: 1–27, 1983.
14. V. Torre and T. A. Poggio, On edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.,* **8**: 147–163, 1986.
15. R. A. Hummel, Representations based on zero-crossings in scale-space, *Proc. IEEE Conf. Comput. Vision Pattern Recognition,* Miami, FL, 1986, pp. 204–209.
16. J. J. Koenderink, The structure of images, *Biol. Cybern.,* **50**: 363–370, 1984.

17. A. P. Witkin, Scale-space filtering, *Proc. Int. J. Conf. Artif. Intell.,* Karlsruhe, 1983, pp. 1019–1022.

18. A. L. Yuille and T. A. Poggio, Scaling theorems for zero-crossings, *IEEE Trans. Pattern Anal. Mach. Intell.,* **8**: 15–25, 1986.

19. K. Shanmugam, F. M. Dickey, and J. A. Green, An optimal frequency domain filter for edge detection in digital pictures, *IEEE Trans. Pattern Anal. Mach. Intell.,* **1**: 37–49, 1979.

20. D. Slepian and H. O. Pollak, Prolate spheroidal wave functions, Fourier analysis and uncertainty—I, *Bell Syst. Tech. J.,* **40**: 43–46, 1961.

21. H. J. Landau and H. O. Pollak, Prolate spheroidal wave functions, Fourier analysis and uncertainty—II, *Bell Syst. Tech. J.,* **40**: 65–80, 1961.

22. J. Canny, *Finding Edges and Lines in Images,* Tech. Rep. 720. Cambridge, MA: MIT Artificial Intelligence Lab., 1983.

23. R. Deriche, Optimal edge detection using recursive filtering, *Proc. 1st Int. Conf. Comput. Vision,* London, 1987, pp. 501–505.

24. J. M. S. Prewitt, Object enhancement and extraction, in B. Lipkin and A. Rosenfeld (eds.), *Picture Processing and Psychopictories,* New York: Academic Press, 1970, pp. 75–149.

25. M. Hueckel, An operator which locates edges in digitized pictures, *J. Assoc. Comput. Mach.,* **18** (1): 113–125, 1971.

26. M. Hueckel, A local visual operator which recognizes edges and lines, *J. Assoc. Comput. Mach.,* **20** (4): 634–647, 1973.

27. I. Abdou, Quantitative methods of edge detection, Ph.D. thesis, University of Southern California, Los Angeles, 1978.

28. R. M. Haralick, Edge and region analysis for digital image data, *J. Comput. Graphics Image Process.,* **12** (1): 60–73, 1980.

29. R. M. Haralick, Digital step edges from zero-crossings of second directional derivatives, *IEEE Trans. Pattern Anal. Mach. Intell.,* **6**: 58–68, 1984.

30. L. T. Watson, T. J. Laffey, and R. M. Haralick, Topographic classification of digital image intensity surfaces using generalized splines and the discrete cosine transform, *J. Comput. Vision, Graphics Image Process.,* **29** (2): 143–167, 1985.

31. V. S. Nalwa and T. O. Binford, On detecting edges, *IEEE Trans. Pattern Anal. Mach. Intell.,* **8**: 699–714, 1986.

32. R. Nevatia and K. R. Babu, Linear feature extraction and description, *J. Comput. Graphics Image Process.,* **13**: 257–269, 1980.

33. A. Huertas and G. G. Medioni, Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks, *IEEE Trans. Pattern Anal. Mach. Intell.,* **8**: 651–664, 1986.

34. H. Freeman, Computer processing of line drawing images, *Surveys,* **6** (1): 57–97, 1974.

35. U. Ramer, An iterative procedure for the polygonal approximation of plane curves, *Comput. Graphics Image Process.,* **1**: 244–256, 1972.

36. H. G. Barrow and R. J. Popplestone, Relational descriptions in picture processing, *Machine Intelligence,* Edinburgh: Edinburgh Univ. Press, 1971, pp. 377–396.

37. T. Pavlidis, *Algorithms for Graphics and Image Processing,* Rockville, MD: Computer Science, 1982.

38. H. Asada and M. Brady, The curvature primal sketch, *IEEE Trans. Pattern Anal. Mach. Intell.,* **8**: 2–14, 1986.

39. F. Mokhtarian and A. K. Mackworth, Scale based description and recognition of planar curves and two-dimensional shapes, *IEEE Trans. Pattern Anal. Mach. Intell.,* **8**: 34–43, 1986.

40. G. G. Medioni and Y. Yasumoto, Corner detection and curve representation using cubic b-splines, *Comput. Vision Graphics Image Process.,* **39** (3): 267–278, 1987.

41. R. Deriche and O. Faugeras, 2-D curve matching using high curvature points: application to stereo vision, *Int. Conf. Pattern Recognition,* Atlantic City, NJ, 1990, pp. 18–23.

42. P. Saint-Marc, H. Rom, and G. G. Medioni, B-spline contour representation and symmetry detection, *IEEE Trans. Pattern Anal. Mach. Intell.,* **15**: 1191–1197, 1993.

43. R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling,* Los Altos, CA: Morgan Kaufmann, 1987.

44. R. O. Duda and P. E. Hart, Use of the Hough transform to detect lines and curves in pictures, *Commun. ACM,* **15** (1): 11–15, 1972.

45. P. V. C. Hough, *Method and means for recognizing complex patterns,* US Patent 3,069,654, 1962.

46. N. Kiryati, Y. Eldar, and A. M. Bruckstein, A probabilistic Hough transform, *Pattern Recognition,* **24**: 303–316, 1991.

47. L. Xu and E. Oja, Randomized Hough transform (RHT): basic mechanisms, algorithms, and computational complexities, *Comput. Vision Graphics Image Process.,* **57** (2): 131–154, 1993.

48. H. Kalviainen et al., Probabilistic and nonprobabilistic Hough transforms: Overview and comparisons, *Image Vision Comput.,* **13** (4): 239–252, 1995.

49. M. Kass, A. P. Witkin, and D. Terzopoulos, Snakes: active contour models, *Int. J. Comput. Vision,* **1** (4): 321–331, 1988.

50. P. Fua and Y. G. Leclerc, Model driven edge detection, *Proc. Image Understanding Workshop,* Cambridge, MA, 1988, pp. 1016–1021.

51. S. Menet, P. Saint-Marc, and G. G. Medioni, B-snakes: implementation and application to stereo, *Proc. Image Understanding Workshop,* Pittsburgh, PA, 1990, pp. 720–726.

52. K. A. Narayanan and A. Rosenfeld, Image smoothing by local use of global information, *IEEE Trans. Syst. Man Cybern.,* **11**: 826–831, 1981.

53. T. N. Pappas and N .S. Jayant, An adaptive clustering algorithm for image segmentation, *Int. Conf. Comput. Vision,* Tarpon Springs, FL, 1988, pp. 310–315.

54. R. Ohlander, K. E. Price, and R. Reddy, Picture segmentation by a recursive region splitting method, *Comput. Graphics Image Process.,* **8**: 313–333, 1978.

55. A. Leonardis, A. Gupta, and R. Bajcsy, Segmentation of range images as the search for geometric parametric models, *Int. J. Comput. Vision,* **14** (3): 253–277, 1995.

56. P. J. Besl and R. C. Jain, Segmentation through variable-order surface fitting, *IEEE Trans. Pattern Anal. Mach. Intell.,* **10**: 167–192, 1988.

57. S. L. Horowitz and T. Pavlidis, Picture segmentation by a tree traversal algorithm, *J. Assoc. Comput. Mach.,* **23** (2): 368–388, 1976.

58. H. Samet, Region representation: Quadtree from binary arrays, *Comput. Graphics Image Process.,* **13** (1): 88–93, 1980.

59. H. Blum, Transformation for extracting new descriptions of shape, *Proceedings of Symposium on Models for the Perception of Speech and Visual Form,* Cambridge, MA: MIT Press, 1967, pp. 362–380.

60. S. M. Pizer, W. R. Oliver, and S. H. Bloomberg, Hierarchial shape description via the multiresolution symmetric axis transform, *IEEE Trans. Pattern Anal. Mach. Intell.,* **9**: 505–511, 1987.

61. A. H. Barr, Superquadrics and angle-preserving transformations, *IEEE Comput. Graphics Appl.,* **1** (1): 11–23, 1981.

62. D. Mumford and J. Shah, Optimal approximations by piecewise smooth functions and variational problems, *Commun. Pure Appl. Math.,* **42** (5): 577–685, 1988.

63. J. M. Morel and S. Solimini, *Variational Methods in Image Segmentation,* Boston: Birkhaeuser, 1995.

64. P. Perona and J. Malik, Scale space and edge detection using anisotropic diffusion, *Proc. IEEE Workshop Comput. Vision, Representation Control,* Miami Beach, FL, 1987, pp. 16–22.

65. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.,* **6**: 721–741, 1984.
66. A. Blake and A. Zisserman, *Visual Reconstruction,* Cambridge, MA: MIT Press, 1987.
67. W. Förstner, A framework for low level feature extraction, *Eur. Conf. Comput. Vision,* Stockholm, Sweden, **2**: 1994, pp. 383–394.

GÉRARD MEDIONI
University of Southern California

FEATURE MAPS, SELF-ORGANIZING. See SELF-ORGANIZING FEATURE MAPS.

FEATURE REMOVAL, HIDDEN. See HIDDEN FEATURE REMOVAL.

FEED ANTENNAS. See REFLECTOR ANTENNAS; ROBUST CONTROL.