COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 51, 256–274 (1990)

# Edge Contours Using Multiple Scales*

DONNA J. WILLIAMS AND MUBARAK SHAH

*Computer Science Department, University of Central Florida, Orlando, Florida 32816*

Received September 16, 1988; revised July 24, 1989

An algorithm for finding a single good path through the set of edge points detected by gradient of Gaussian operator is discussed. First, an algorithm for finding contours at one scale is presented, then an extension of that algorithm which uses multiple scales to produce improved detection of weak edges is presented. The set of possible edge points is placed on a priority queue with the edge point having largest magnitude at the top. The strongest edge point that is not already on a contour is retrieved from the queue. The point in the computed direction is examined first, then in those in the adjacent directions on either side of it. Each branch is followed to the end and a weight assigned at each point based on four factors: a measure of noisiness, a measure of curvature, contour length, and the gradient magnitude. The point with the largest average weight is chosen. After searching from the initial point in one direction, a similar search is conducted in the oppositedirection unless a closed contour has been formed. In the algorithm for multiple scales the search for a contour proceeds as for the single scale, using the largest scale, until a best partial contour at that scale has been found. Then the next finer scale is chosen and the neighborhood around the end points of the contour are examined to determine possible edge points in a direction similar to the end point of the contour. The original algorithm is then followed for each of the points satisfying the above condition, and the best is chosen as an extension to the original edge. Further, in order to determine the size neighborhood that should be searched when attempting to pick up an edge at a smaller scale, a theoretical analysis of the movement of idealized edges is performed. This analysis examines two adjacent step edges having the same parity (a staircase) and opposite parity (a pulse). It is determined that maximum movement for both cases is $\sigma$, where $\sigma$ is the standard deviation of the Gaussian used. This maximum movement occurs for the staircase when the two nearby edges have the same step size and are at a distance of $2\sigma$ apart. However, for edges closer or farther away, maximum movement decreases rapidly. For a pulse, maximum movement occurs when the two edges have the same step size and are very close together. Again the movement decreases rapidly as the edges become farther apart. Movement also decreases in both cases when the relative strengths of the two edges are not equal 

## I. INTRODUCTION

The field of computer vision is concerned with extracting the information contained in images about scenes they depict. The effectiveness of the early levels of processing which identify tokens to be used at the higher levels of processing is crucial in determining how successful this higher level processing will be.

Edge point detectors identify potential edge points, those where the image intensity is changing rapidly. These typically return an edge map identifying the location of points where the intensity gradient is high, together with some gradient and direction information. In order to use this information in higher level processing, the next step is to identify those points which should be grouped together into edge segments. Several authors have developed algorithms for linking edge points

*The research reported in this paper was supported by the Center for Research in Electro Optics and Lasers (CREOL) University of Central Florida under Grant 20-52-043 and by the National Science Foundation under Grant IRI 87-13120.



0734-189X/90 $3.00

into segments. Martelli [12] used a heuristic search algorithm which required beginning and end points and a method of estimating the cost of the path from any point to the end point. Fischler and Wolf [8] used a minimum spanning tree approach. Nevatia and Babu [13] first applied directional $5 \times 5$ gradient masks, then performed nonmaxima suppression to identify edge points. The direction of a point was compared to those of its neighbors to determine its successors and predecessors in an edge. Then, beginning with points having no predecessors, the points were linked. When a point had two successors, the primary one, based on proximity and magnitude, was linked first. The secondary successor was considered the beginning of a separate edge segment. Thus the choice of which points to link was a local decision.

The choice of the size neighborhood, or scale, to use in smoothing an image and determining the gradient is a difficult problem. When using the Gaussian function to smooth images, $\sigma$, the standard deviation of the Gaussian, is the scale parameter. Smaller scales result in too much noise and fine texture while larger scales result in delocalization of edges. The use of multiple scales in edge detection to reduce the conflicting goals of noise suppression and accurate localization goes back as far as Rosenfeld and Thurston [15]. Based on psychophysical findings, Marr and Hildreth [11] advocated the use of multiple scales defined using different standard deviations of the Gaussian. Leipnik [10] and Torre and Poggio [18] showed that the Gaussian is optimal for reducing noise with minimum delocalization. A number of authors have proposed methods to combine the information obtained at different scales. Among them are Marr and Hildreth [11], Eklundh, Elfving, and Nyberg [7], Canny [4], Schunck [16], and Bischof and Caelli [3].

Witkin [21] introduced the concept of *scale space*, where the zero crossings of the second derivative are examined for a continuous spectrum of $\sigma$ values rather than only for a small number of discrete values. The properties of scale space have been examined by a number of authors. Clark [5, 6] examined scale space from the perspective of catastrophe theory. Shah, Sood, and Jain [17] analyzed the qualitative nature of the scale space of staircase and pulse edges, as did Katz [9]. Peich [14] showed that the scale space of the two-dimensional staircase and pulse have a cross section identical to that of the one-dimensional edge models. Thus it is reasonable to analyze one-dimensional models to determine behavior of edges which are not near corners. Berzins [1] and Bergholm [2] examined the behavior of corners and small closed curves in scale space. Bergholm also analyzed the amount of movement exhibited by a pulse composed of two *equal* step edges having opposite parity. He used this information to determine the size of the smallest neighborhood of an edge point detected at one scale of Gaussian that would be assured to contain that same edge at a different scale. Using a technique called *edge focusing* he produced an edge map identifying the edge points at the finest scale that could be tracked from a coarse scale edge. This removed the effects of delocalization and over smoothing that occurred at larger scales. However, for diffuse edges, e.g., shadows, the edge was replaced by noise and fine texture edges that were nearby.

This paper presents a method of producing connected edge contours which are suitable for higher processing. The algorithm first uses a gradient of Gaussian operator to determine gradient magnitude and direction, followed by a nonmaxima suppression step to identify ridges in the gradient map. Canny [4] has shown that this is a near optimal edge detector. However, this edge detector, as most others, gives only an edge map identifying points which are gradient maxima. Another problem is that the ridge is often more than one pixel wide, especially for diagonal edge segments, and may have small noisy spurs. Grouping the points which belong to a single edge into a contour only one pixel wide gives a more meaningful data structure. Thus the gradient maxima points are further thinned to one pixel wide and linked using an algorithm which assigns weights based on several factors and then chooses the set of points giving the largest average weight. The weight at each point is computed using local direction and magnitude information and a global length weight. The set of points having the highest average weight is chosen, so global information is used here also. This is in contrast to Nevatia and Babu who used only information in the $3 \times 3$ neighborhood of a point to determine which point to link to it. This single scale edge linking algorithm is then extended to one which uses multiple scales to obtain better edges with little increase in the response to noise.

In order to link together edge points which have been detected at different scales it is necessary to know how far from the original position or how far from the position at another scale an edge point will be detected. Consequently we analyze the movement of edge points as they are smoothed with a Gaussian operator at different scales. The edge model chosen for the analysis is the step edge, and combinations examined are the staircase (adjacent edges having the same parity) and the pulse (opposite parity). The ratio of the two stepsizes is arbitrary. Shah *et al.* developed equations for these step pairs convolved with the Gaussian and its derivatives and showed the general shape of the scale space curves. That work is extended to develop the equation of the scale space curves and analyze quantitatively the amount of the delocalization that occurs as images containing these steps are convolved with Gaussians having different values of $\sigma$. In some cases the edge location approaches a certain limiting position. The equations for these positions are also developed. Bergholm examined a pulse having equal steps and showed that the speed with which two edges move apart is limited by $\Delta\sigma$. This paper considers general pulses as well as staircases and focuses on the maximum possible movement as a function of $\sigma$. The step edge was chosen because of its ease of analysis and because it exhibits the most extreme delocalization, thus giving an upper bound on the distance an edge can move.

## 2. SINGLE SCALE EDGE DETECTION AND LINKING

In this section, we present an algorithm for finding a single good path through the set of gradient maximum points. In this method, the image is first convolved with a gradient of Gaussian operator. The two-dimensional Gaussian with standard deviation $\sigma$ is defined by the function

$$g(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

This can be separated into the product of two one-dimensional functions

$$g(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$

The derivative is $g'(x) = -(x/\sigma^2)g(x)$. Convolving the image $f(x, y)$ in the horizontal direction with $g'(x)$, then in the vertical direction with $g(y)$ gives the partial derivative of $f$ with respect to $x$, $f_x$, while at the same time smoothing the image in both directions with the Gaussian. The degree of smoothing depends on the value of $\sigma$ used, larger values giving more smoothing. Similarly, convolving with $g'(y)$ in the vertical direction followed by $g(x)$ in the horizontal direction gives the partial of $f$ with respect to $y$, $f_y$. These are the components of the gradient vector $(f_x, f_y)$, which gives the rate of change of intensity in the $x$ and $y$ directions at each point in a smoothed image. The magnitude of the rate of change is $M(x, y) = \sqrt{f_x^2 + f_y^2}$. The direction of maximum rate of change at each point can be computed to be $\theta = \tan^{-1}(f_y/f_x)$, $0 \leq \theta < 360°$. $M(x, y)$ is also equivalent to the directional derivative of $f$ in the direction of greatest change.

The set of possible edge points is placed in a priority queue with the edge point having largest magnitude on the top. Thus the strongest edge points will be extended into contours first. This step is not critical, and the edge point to extend could be determined by a simple left to right scan. However, in some cases the order in which the points are processed could make a difference. For example, if three edge segments met at a point, the segment having the strongest edge points would consider the junction first and decide which of the two segments to join. The third segment would become a separate contour. This situation rarely occurs except in the case where one of the three segments is a noisy spur, because the nonmaxima suppression step usually causes a gap between the main contour and a side branch.

The search for points to assign to a contour proceeds as follows. The first edge point that is not already on a contour is retrieved from the queue. The gradient values are used to determine the direction $\theta + 90°$ of the next edge point. This assumes that the edge will be at a right angle to the direction of greatest intensity change. The angle is converted into an integer 0 through 7, each representing a 45° range. Zero corresponds to an edge in the range $-22.5°$ to $22.5°$, with the subsequent integers going counterclockwise from 0. The point in the computed direction is examined first, then those in the adjacent directions on either side of it. Each branch is followed to the end and a weight assigned at each point based on four factors. The four factors are

1. Is the edge point being examined in the direction determined by the gradient, or in the direction next to it?
2. How much does the direction of the next edge point differ from that of the current edge point?
3. What is the magnitude of the gradient?
4. Contour length.

The maximum weight assigned is 40 with each of the four factors contributing a maximum of 10 each. First, if the point being examined is in the direction pointed to by the previous point, the weight is 10. Otherwise it is 45° on one side or the other and the weight is 5. In Fig. 1a the previous point, $p$, with its direction is on the left. The three points that are possible successors are on the right together with their weights for factor one. For the second factor, the direction of the

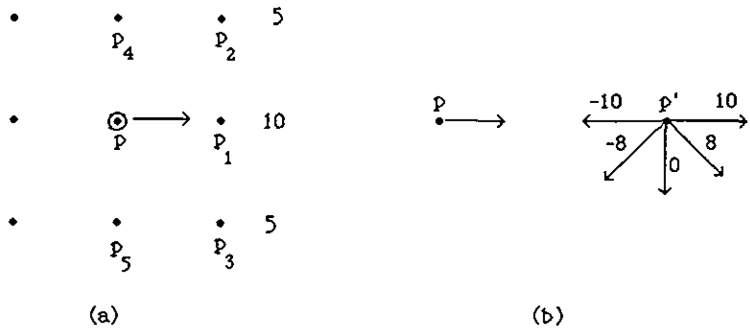



FIG. 1. (a) Weights for factor one. Previous point, $p$, is on the left and possible successors, $p_1, p_2, p_3$, with their weights on the right. (b) Weights for factor two. Previous point, $p$, with its direction on the left and current point $p'$ on the right with weights determined by comparing its direction with that of $p$. Ten points for a difference of 0; 8 for a difference of 1; 0, $-9$, $-10$ for differences of 2, 3, 4, respectively.

current point, $p'$, is compared to the direction of the previous point, $p$. Ten is assigned for a difference of 0, 8 for a difference of 1, 0 for a difference of 2, $-8$ and $-10$ for a difference of 3 and 4, respectively. In Fig. 1b the previous point with its direction is on the left. The current point, with weights for possible directions is on the right. There is little penalty for a difference of 1, as this often occurs on curves and lines near the boundary of the direction regions (e.g., 20–25°). However, sharp corners are penalized. This factor is especially important at the end of a contour where the location of an edge is more inclined to drift. The third weight is based on gradient magnitude and is the ratio of the point's magnitude to the maximum magnitude in the image multiplied by 10. The fourth factor is designed to penalize short edges and is equal to the length of the edge constructed from this point to the end, if the length is less than 10. Otherwise the weight is 10.

The search space is reduced by the nonmaxima suppression step, so there are a limited number of possible paths through the points. The weights are designed to favor the longest, strongest, straightest path. After points in the three primary directions have been examined, the path having the largest average weight is chosen and returned to the calling program. The traversal of a branch terminates when there is no potential edge point in any of the three primary directions.

After searching from the initial point in one direction, a similar search is conducted in the opposite direction, unless a closed contour has been formed. The two branches are combined to form one contour. Contours having three or fewer pixels are discarded. Then the next point is chosen from the queue, and the search continues for the next contour until there are no more edges in the queue having magnitude greater than a threshold expressed as a percent of the maximum magnitude. Values in the range 5 to 10% gave good results on the images presented here.

## 3. MULTIPLE SCALES

This section deals with the extension of the algorithm presented in the last section to an algorithm which uses multiple scales in order to produce improved detection of weak edges. It is known that, at higher scales, edges are delocalized. In our algorithm for multiple scales we need to know how large the delocalization is. We have done theoretical analysis of the movement of idealized edges undergoing Gaussian smoothing which is presented in Section 3.1. The algorithm for multiple scales is discussed in Section 3.2.

### 3.1. *Maximum Movement of Edges*

It is well known that smoothing an image with a Gaussian operator causes delocalization of edges. In this section the question of how far an edge can move will be analyzed. The edge model used will be the ideal step edge, and the two cases considered will be neighboring edges having the same and opposite parity. These will be referred to as the *staircase* and the *pulse* edge types. The unit step edge is represented by the equation

$$U(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise.} \end{cases}$$

The staircase with edges located at $-a$ and $a$ is represented by

$$S_a(x) = bU(x + a) + U(x - a),$$

while the pulse with edges at $-a$ and $a$ is represented by

$$P_a(x) = bU(x + a) - U(x - a).$$

See Fig. 2. After convolving with the derivative of the Gaussian, the equation for the staircase is

$$s_{\sigma,a}(x) = bg(x + a) + g(x - a) \tag{1}$$

and that for the pulse is

$$p_{\sigma,a}(x) = bg(x + a) - g(x - a). \tag{2}$$

It will be assumed that $b$, which represents the relative heights of the two steps, satisfies $0 < b \leq 1$. Thus we are considering the weaker edge to be at $x = -a$.

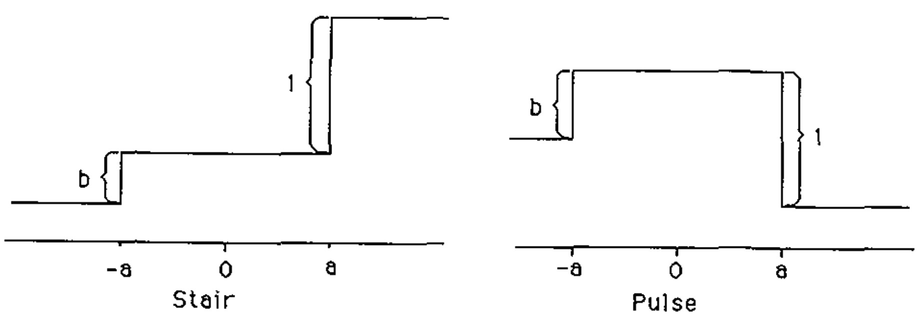



FIG. 2. Staircase and pulse edges.

The method used in this work uses the maxima of the gradient magnitude to detect edges. The equation of $\sigma$ as a function of $x$ at the maxima for the staircase is

$$\sigma \approx \left[\frac{2ax}{\ln(b(a+x)/(a-x))}\right]^{1/2}$$

and for the pulse

$$\sigma = \left[\frac{2ax}{\ln(b(x+a)/(x-a))}\right]^{1/2}.$$

These equations are derived by setting the derivatives of Eqs. (1) and (2) equal to 0, then rearranging terms. For more detail see [19]. The graphs for several values of $b$ are given in Figs. 3 and 4. The middle branch which appears for small values of $\sigma$ on all the graphs represents a gradient minimum rather than a maximum, so it does not correspond to an edge. These will be referred to as the scale space images. For a more complete discussion on the scale space images of pulse and staircase edge pairs see [9, 17].

First we will consider the staircase. Notice that if $b = 1$ the edges move together until they meet when $\sigma = a$, then only one edge exists at $x = 0$ for $\sigma > a$. When $b < 1$ the stronger edge moves toward the middle and approaches the asymptote $a(1 - b)/(1 + b)$ as $\sigma$ approaches $\infty$. The asymptote is determined by setting the derivative of Eq. (1) to 0, combining terms having an $x$ factor, then taking the limit as $\sigma \to \infty$. When $b < 1$, the maximum corresponding to the weak edge on the left disappears when $\sigma$ becomes sufficiently large. Thus when $b = 0.8$ the maximum movement of the weak edge occurs just before it disappears and is $(1 - 0.493)a = 0.507a$. When $b = 0.3$ the maximum movement is $(1 - 0.723)a = 0.277a$. The units on both axes in the scale space image are marked in units of $a$. This can be done because if $x$ and $\sigma$ are both multiplied by $a$, then $s_{a\sigma,a}(ax) = bg_{a\sigma}(ax + a) + g_{a\sigma}(ax - a)$. But

$$\begin{aligned} g_{a\sigma}(ax \pm a) &= 1/\sqrt{2\pi}\,a\sigma \exp\left(-(ax \pm a)^2/2a^2\sigma^2\right) \\ &= 1/\sqrt{2\pi}\,a\sigma \exp\left(-(x \pm 1)^2/2\sigma^2\right) = (1/a)g_\sigma(x \pm 1). \end{aligned}$$

Thus $s_{a\sigma,a}(ax) = (1/a)s_{\sigma,1}(x)$; the function has been multiplied by a constant, but the location of the maxima will occur at the same locations, and one graph can be used to represent all values of $a$.

In practice, when an image is being examined, $\sigma$ is known, but $a$ is not. Thus, instead of considering $a$ constant and $\sigma$ as the variable in Figs. 3 and 4, $\sigma$ can be fixed and $a$ can be allowed to vary. For a fixed value of $\sigma$, different points on the vertical axis will then correspond to different values of $a$. In this discussion, whenever a fixed value of $\sigma$ is being considered, $\bar{\sigma}$ will be used instead of $\sigma$ to indicate this.

For example, if $\sigma = 2$, the point $2a$ on the vertical axis will correspond to $a = 1$, ($\sigma = 2 = 2a$, thus $a = 1$), while the point $a$ will correspond to $a = 2$ and $0.5a$ will correspond to $a = 4$. In general, if $\hat{\sigma}a$ is a point on the $\sigma$ axis, then $\bar{\sigma} = \hat{\sigma}a$,

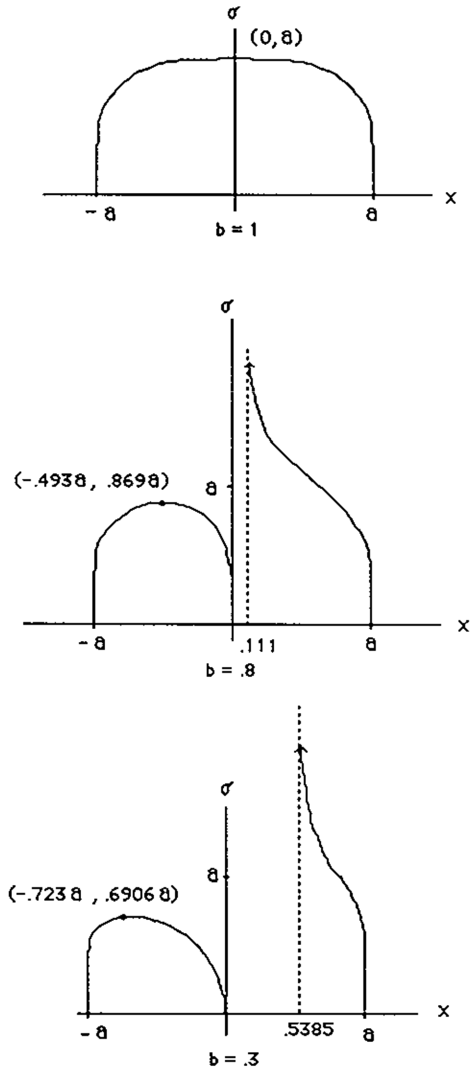



FIG. 3. Scale space image: Location of gradient extrema for staircase when (a) $b = 1$, (b) $b = 0.8$ (c) $b = 0.3$.

$a = (1/\hat{\sigma})\bar{\sigma}$. Thus for fixed $\sigma$, larger points on the vertical axis correspond to smaller values of $a$, while smaller points correspond to larger $a$. If $(x, \hat{\sigma})$ are the coefficients of a point on the scale space curve, the distance that the edge point has moved from its original location at $a$ will be $(1 - x)a = (1 - x)(1/\hat{\sigma})\bar{\sigma}$ for the strong edge and $(1 + x)a = (1 + x)(1/\hat{\sigma})\bar{\sigma}$ for the weaker edge.

Since movement $(m)$ depends on $a$, and both $m$ and $a$ can be expressed in terms of $\bar{\sigma}$, this suggests plotting $m$ versus $a$. This is shown in Fig. 5. On both axes the units are $\bar{\sigma}$. Notice that when $b = 1$, if two edges are $2\bar{\sigma}$ apart $(a = \bar{\sigma})$ each will move $\bar{\sigma}$ and combine to become one edge. If the edges are closer together

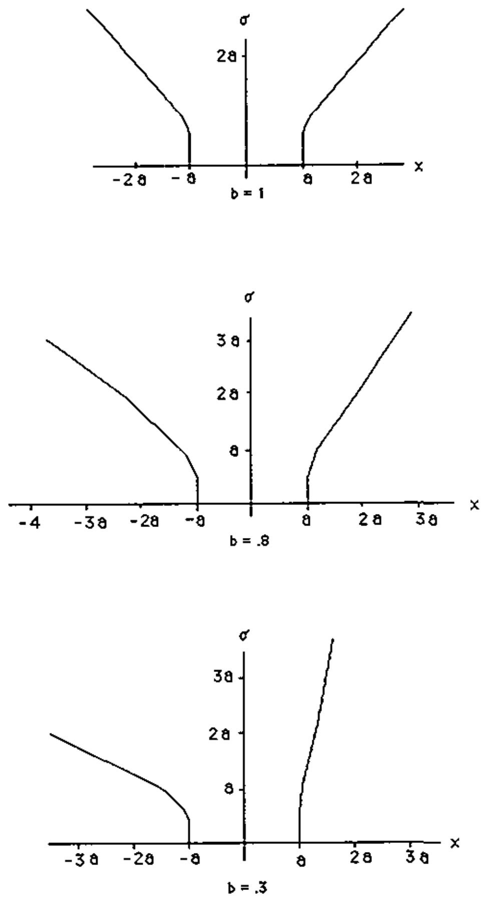



FIG. 4. Scale space image: Location of gradient maxima for pulse when (a) $b = 1$, (b) $b = 0.8$, (c) $b = 0.3$.

(smaller value of $a$), they will move $a$ pixels each to come together, but because they were closer to begin with, the distance moved will be smaller. Thus for $a < \bar{\sigma}$, $m = a$. If the distance apart is greater than $2\bar{\sigma}$ ($a > \bar{\sigma}$) they will move closer together, but remain distinct. When $a > 2\bar{\sigma}$ the amount of movement will be negligible. This corresponds to the part of the scale space image where the curve is nearly vertical. In this situation the edges are far enough apart to have little interaction, so there is little movement. This is a result of the fact that 99% of the support of a Gaussian filter having standard deviation $\sigma$ falls within $3\sigma$ of the mean, so there is effectively no interaction when edges are this far apart. The interaction begins slowly as the edges become closer together. When $a = 2\bar{\sigma}$, the movement is only about $0.0014\bar{\sigma}$, or less than 0.1% of the distance from the edge to the center point. As $a$ becomes smaller, the movement increases rapidly to the maximum at $(\bar{\sigma}, \bar{\sigma})$, then decreases until $a$ reaches 0.

For $b < 1$, the movement of the stronger edge will be largest for some value of $a$ between 0 and $\bar{\sigma}$. Figure 6 shows a graph of the maximum movement possible, in terms of $\bar{\sigma}$, for different values of $b$, for the stronger and weaker edges in the

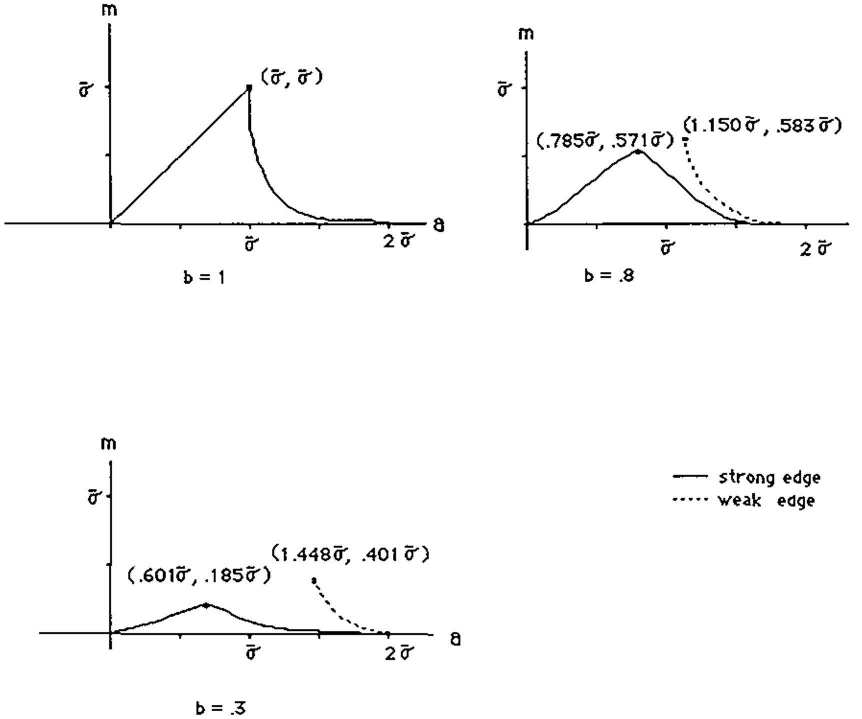



FIG. 5. Movement vs distance between edges, staircase.

staircase. Always the largest movement of $\sigma = \bar{\sigma}$ will occur for equal edges which are $2\bar{\sigma}$ apart. For example, if an image is convolved with the gradient of Gaussian with $\sigma = 2$, then maximum movement is two pixels and occurs when $b = 1$ and $a = 2$. However, if it were known that most neighboring edges had relative strength 0.5, their greatest movement would be about 0.92, or less than one pixel. Similarly, edge pairs that had separations less or greater than four pixels would move less than two pixels.

A similar analysis can be performed for a pulse. Figure 7 shows the movement versus the distance between edges for a pulse. The maximum movement, when $b = 1$, is $\bar{\sigma}$, as it was for the staircase, but this value is now the limiting value as

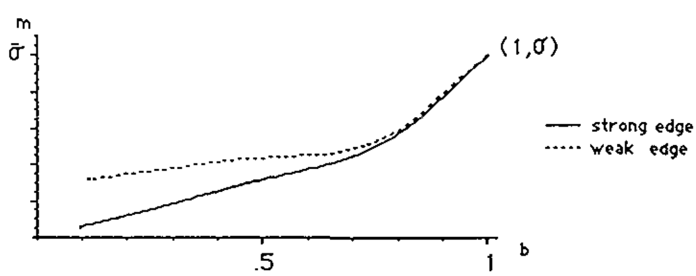



FIG. 6. Maximum movement in terms of $\bar{\sigma}$ vs $b$ for weaker and stronger edges in a staircase.

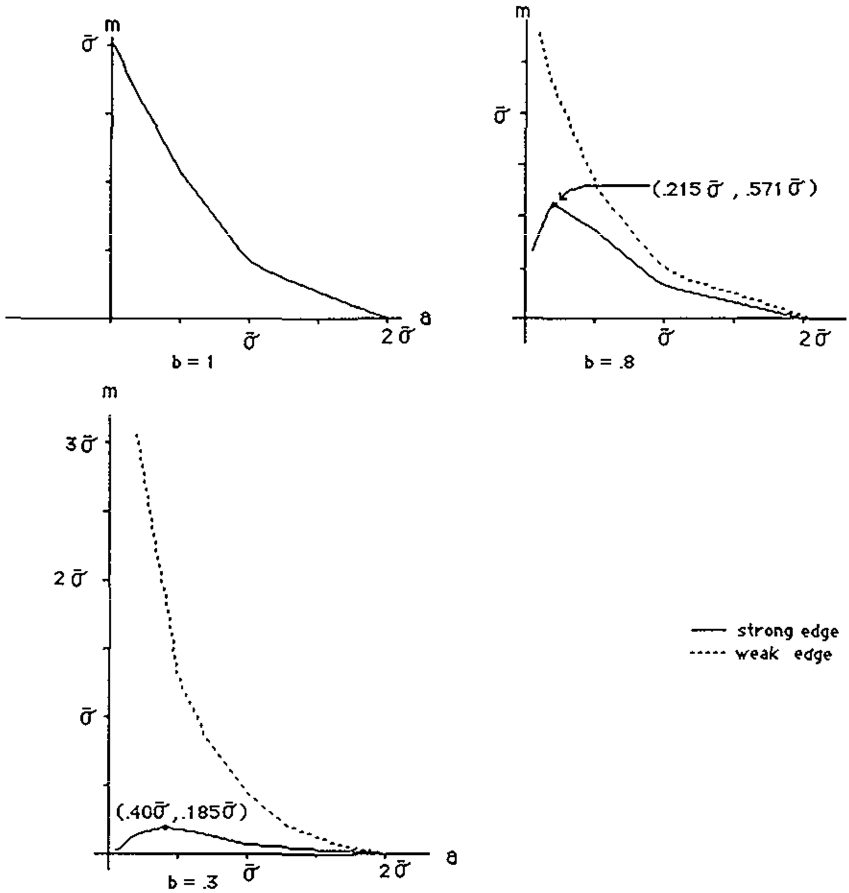



FIG. 7. Movement vs distance between edges, pulse.

the edges become closer together. This can be seen by examining the scale space graphs for the pulse (Fig. 4). When $b = 1$ the curve approaches the lines $\sigma = \pm x$. Movement is $(|x| - 1)a = (|x| - 1)\bar{\sigma}/\hat{\sigma}$. Since the curve approaches the lines $\sigma = \pm x$, the points on the curve approach $(\sigma, \sigma) = (x, x)$ for large values of $\sigma$ and $x$. So $\lim_{x\to\infty} m = \lim_{x\to\infty}((x - 1)/\hat{\sigma})\bar{\sigma} = \lim_{x\to\infty}((x - 1)/x)\bar{\sigma} = \bar{\sigma}$. But as $x$ goes to infinity, for $\sigma = \bar{\sigma}$, $a$ goes to 0. Thus $\lim_{a\to 0} m = \bar{\sigma}$.

When $b < 1$, the strong edge in the scale space image approaches the vertical asymptote $a(1 + b)/(1 - b)$ and displays a well-defined maximum movement as in the staircase, for a value of $a$ between 0 and $\bar{\sigma}$. But the weak edge can move indefinitely as $a$ becomes smaller. In the scale space image the weak edge approaches the horizontal parabola $x = (\ln b/2a)\sigma^2$. This asymptote for the weaker edge is obtained by rearranging the equation for the scale space of a pulse to get

$$\frac{x}{\sigma^2} = \frac{1}{2a}\ln\frac{b(x + a)}{x - a}. \tag{3}$$

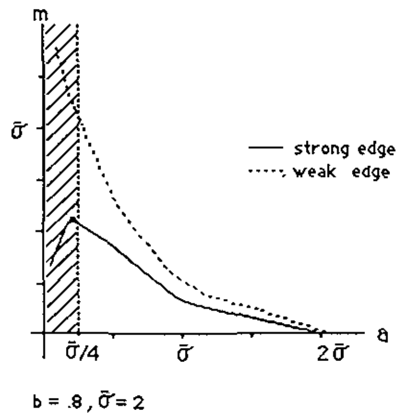



FIG. 8. Maximum movement when $b = 0.8$ and $\bar{\sigma} = 2$.

Then as $x \to -\infty$, $x/\sigma^2 \to (\ln b/2a)$. Thus the curve approaches the parabola $x = (\ln b/2a)\sigma^2$. But $p_{\sigma,a}((\ln b/2a)\sigma^2) = 0$ for all values of $\sigma$, thus the parabola gives the location where the gradient value crosses zero as it goes from the positive step of the pulse to the negative. The rate of change of position as $\sigma$ changes is near that of the parabola, $dx/d\sigma = (\ln b/a)\sigma$, thus it will be unbounded as $a \to 0$ or $b \to 0$.

There are two practical considerations limiting the amount of movement of the weak edge. The first is that as $a$ becomes smaller and the weak edge is moved farther, its location becomes closer to the parabola above, thus its gradient value becomes smaller and, at some point, falls below any threshold being used. The other limiting factor relates to the sampling theorem. In the case of a pulse the wavelength is $4a$. The distance unit being used is the interpixel distance which is assumed to be the same as the sampling distance in the original image. The sampling theorem states that the sampling interval $\delta$ should be less than $\lambda/2$, where $\lambda$ is the wavelength of the highest frequency. Since $\delta$ is 1, $\delta < \lambda/2$ means $2 < \lambda = 4a$. Therefore $a > \frac{1}{2}$. Thus, for $0 < a < \frac{1}{2} = (1/2\bar{\sigma})\bar{\sigma}$, the conditions of the sampling theorem are not met. Figure 8 gives an example when $b = 0.8$ and $\bar{\sigma} = 2$. Then $a = (1/2\bar{\sigma})\bar{\sigma} = (1/4)\bar{\sigma}$ is the cutoff point and maximum movement is $0.5675\bar{\sigma}$ for the strong edge and $1.004\bar{\sigma}$ for the weak edge.

Figure 9 gives a graph of maximum movement for the stronger edge of a pulse as $b$ varies.

In summary, maximum edge movement as a function of $\sigma$ using the gradient of Gaussian can be characterized as follows:

1. Straight, isolated edges—no movement.
2. Two edges not near a corner
   a. Equal parity (staircase). Greatest movement is $\sigma$ and occurs when edges are $2\sigma$ apart and have equal contrast. Movement decreases rapidly for edges closer or farther away and those having unequal stepsize.
   b. Opposite parity (pulse). Maximum movement is $\sigma$ for equal edges and for the stronger of two unequal edges and occurs when edges are very

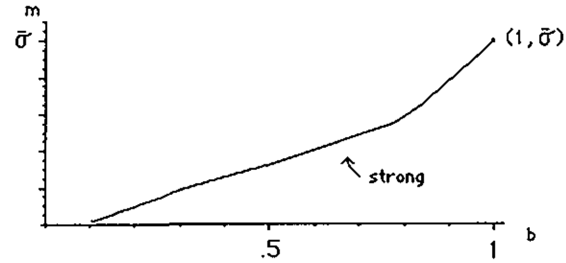



FIG. 9. Maximum movement in terms of $\bar{\sigma}$ vs $b$ for stronger edge in a pulse.

close together. The weaker of two edges can exhibit unbounded movement, but gradient magnitude decreases, so edge will usually not be detected farther than $\sigma$ from its original location.

3. For a step edge which has a corner with angle $> \pi/8$, the edge contour will shrink inside the corner. Displacement will be less than $2\sigma$, decreasing as the angle increases [2].

4. For small closed curves the edge moves out at a speed less than $\Delta\sigma$, approaching a circle of radius $\sigma$ [2].

5. Behavior of corners and small closed curves under the Laplacian of Gaussian operator behave somewhat differently [1].
   a. At a corner, the edge goes through the corner, but bulges outward on both sides a distance less than $\sigma$ for angles $> \pi/12$.
   b. Squares and circles having side or radius less than $4\sigma$ approach a circle of radius $\sqrt{2}\,\sigma$.

The numerical calculations presented in this section were performed on a Macintosh SE computer using Borland's *Eureka* package to solve equations and find maxima and minima.

### 3.2. *Multiscale Algorithm*

This section extends the algorithm given in the previous section to one using multiple scales, as follows. Initially the image is convolved with gradient masks at three scales: $\sigma$, $\sqrt{2}\,\sigma$, and $2\sigma$, where $\sigma$ is input by the user. The search for a contour proceeds as for the single scale, using the largest scale, until a best partial contour at that scale has been found. Then the next finer scale is chosen and the neighborhood around the end points of the contour is examined to determine if there are possible edge points at that scale having a direction similar to the end point of the contour. In this case a difference of two in the directions is considered close enough to continue the edge contour at the smaller scale. This is because the direction may change slightly at a different scale due to the fact that there is less interaction between edges at smaller scales. The neighborhood searched is only one pixel in each direction, based on the analysis in Section 3.1. This analysis shows that the maximum delocalization of an edge point is $\sigma$ and is usually less than that. Thus an edge detected at one scale, $\sigma_1$, should appear no further away than $|\sigma_1 - \sigma_2|$ when the scale is $\sigma_2$. When the largest value of $\sigma$ is 4, most edges will be found in a neighborhood with radius one pixel, and when the largest value is 2, all will be in this neighborhood. The original algorithm is then followed for each of the points satisfying the above condition, and the best is chosen as an extension to the original edge. While extending the edge, if any point is discovered to be a possible edge point at a coarser scale, the search scale is increased to that value.

When the contour cannot be extended further, the scale is decreased to the next finer scale, and the process is repeated until the contour cannot be extended at the finest scale. The edge segment is then extended in the same manner at the other end. This algorithm resulted in a considerable improvement in the detection of some of the incomplete edge contours, with almost no degradation due to inclusion of noisy edge points. Another multiscale method that improves localization is described in [20].

## 4. EXPERIMENTAL RESULTS

The algorithms were tested on several real images. The values of $\sigma$ used were 1, $\sqrt{2}$, and 2 and a threshold of 0.08 was applied. For comparison, the images were also processed using the Canny operator. The threshold used for the Canny algorithm was chosen to return approximately the same number of contours as the edge linking algorithms. The results for three images, Part, Tiwanaku, and Bananasplit, are shown in Figs. 10, 11, and 12. The original image is (a), the Canny operator is (b), the single scale algorithm is (c), and the multiple scale algorithm is (d).

In the Part image, there were several fairly well-defined edges, but quite close to each other, so that at scales which were large enough to remove noise, the nearness of the edges had caused some of them to disappear. The main difference between the Canny operator and the single scale edge-linking algorithm was that the double edges were replaced by single pixel edges, and some small spurs on the edges were eliminated. More improvement was achieved with the multiple scale algorithm. Looking at the fourth contour from the center, the multiple scale algorithm was able to join three partial contours and extend the right side into an almost complete contour. The third contour was also extended across the bottom of the image. Note also, in the shadow edges at the top and bottom of the image, that small fragmentary edge contours have been combined into longer, much more well-defined contours.

In the Tiwanaku image the single scale algorithm shows less noise than the Canny, probably due to the difference in the thresholds. Again, some spurs and double edges have been removed. The multiple scale algorithm produces slight extensions of the vertical lines on the large stone, and the bases of the smaller stones are more well defined. Some of the texture edges in the background and the clouds have been combined into longer contours as well.

The Bananasplit image is different in that most edges fall into one of two categories. They are very well defined, as in the sides of the post and the floor–wall joint that are well marked by all the algorithms, or they are very poorly defined, with few intermediate edges of the type most improved by the multiple scale algorithm. For example, the specularity on the left side of the stool interrupts the bottom edge and the gap cannot be detected at any scale. The horizontal lines on the stool are only one pixel wide, thus the edges are too close together to be

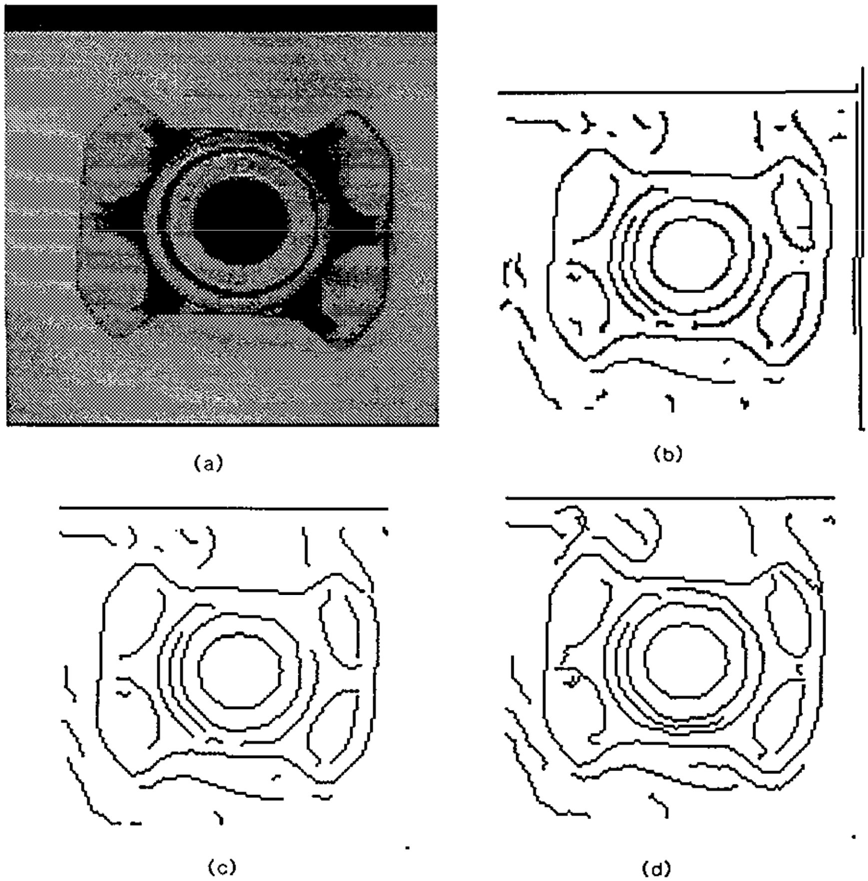
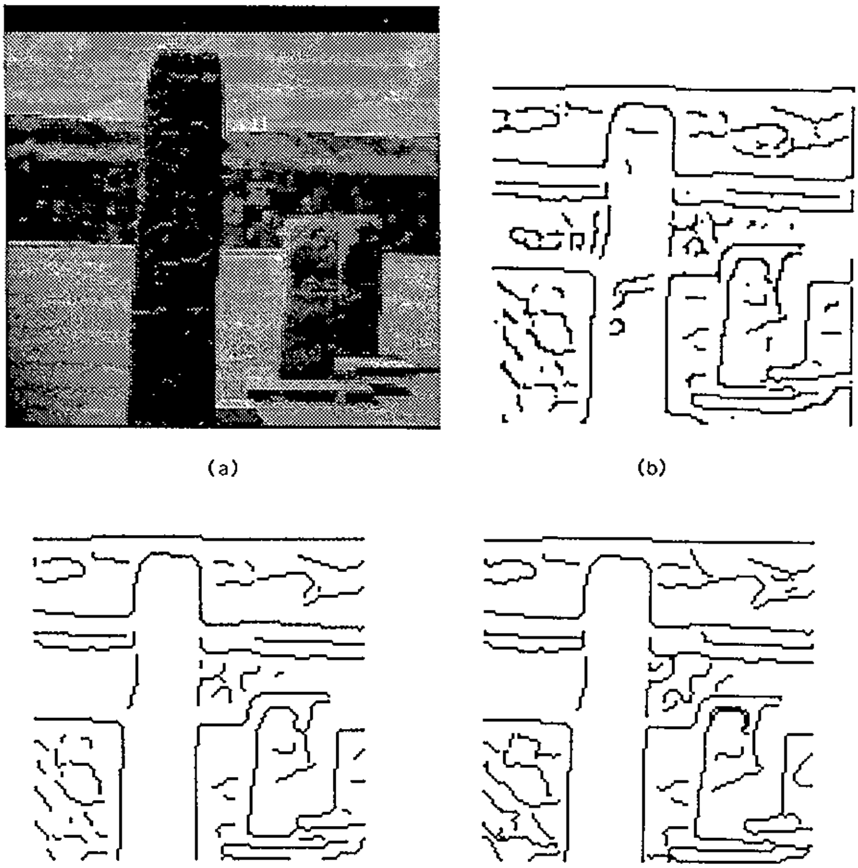


Fig. 10. Part: (a) original image; (b) Canny operator; (c) single scale edge linking algorithm (d) multiple scale algorithm.

detected by any of the methods. The multiple scale algorithm did complete one contour on the left.

The conclusion is that the single scale edge-linking algorithm cleans up the Canny edges and produces a set of linked lists corresponding to the contours found. The multiple scale algorithm is able to improve detection of edges that are close together and interact at scales which are large enough to remove noise and fine texture. It also improves detection of weak, but well-defined edges, such as those of the shadows in the Part image.

The weights were chosen heuristically. Experiments varying the weights indicated that the actual values did not seem to be extremely critical as long as higher weights were given to the points in the primary direction having the same direction as the current point, high magnitude, and longer length contour. Experiments in which each one of the factors in turn was removed, however, indicated that no three gave as good results as using all four. This was interesting especially in relation to weights 1 and 2, which were both determined using direction information. The weight for factor 1 is higher when the point is noise free and the



(a) (b) (c) (d)

FIG. 11. Tiwanaku: (a) original image; (b) Canny operator; (c) single scale edge linking algorithm; (d) multiple scale algorithm.

curvature is small, but noise seems to be the most important factor. Factor 2 has a higher weight when the curve has no sharp turns. Removing the length factor allowed one or two strong points, perhaps lying next to the main edge on a ridge two pixels wide to be chosen, rather than a longer contour which extended into a weaker portion of the edge.

## APPENDIX

SINGLE SCALE ALGORITHM.

procedure FIND CONTOUR

1. get next POSSIBLE EDGE point, $p$
2. FOLLOWEDGE($p, C, wt, len$) in forward direction
3. FOLLOWEDGE($p, D, wt, len$) in backward direction
4. COMBINE($C$, $D$) into a single contour

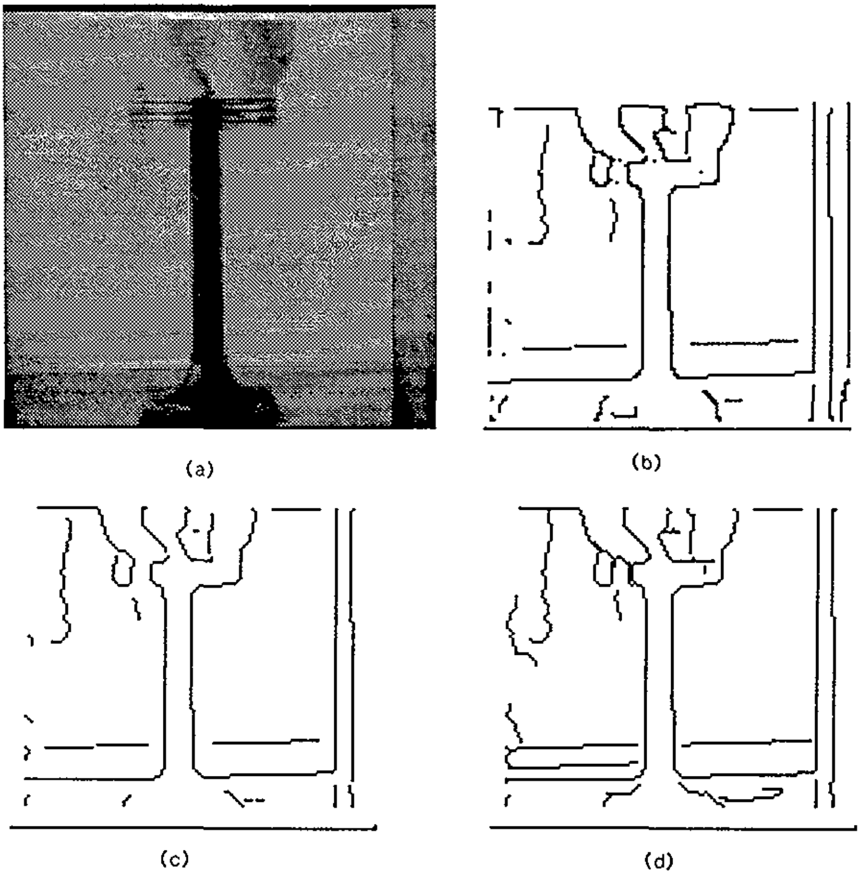



FIG. 12. Bananasplit: (a) original image; (b) Canny operator; (c) single scale edge linking algorithm; (d) multiple scale algorithm.

procedure FOLLOWEDGE($p, C, wt, len$)
input: $p$, point to begin contour
output: $C$, contour points
$wt$, average weight of $C$
$len$, number of pixels in $C$

1. **if not** POSSIBLE EDGE($p$) **then** $wt = len = 0$, **return**
2. **for** $i = 1, 2, 3$ **do** FOLLOWEDGE($p_i, C_i, wt_i, len_i$)/* see Fig. 1a for numbering of points */
3. $max = i$, where $wt_{max}$ is the largest of $wt_i$
4. **if** $wt_{max} = 0$ **then** /* $p$ has no continuation */
   $wt = factor1\_wt + factor2\_wt + factor3\_wt + 1$
   $len = 1$
   $C$ = PUSH($\Phi, p$) /* Create a new contour $C$ containing $p$ */
   **return**
5. /* continuation found */
   $wt = (len_{max} * (wt_{max} - len_{max}) + factor1\_wt + factor2\_wt + factor3\_wt)/(len_{max} + 1) + factor4\_wt$

6. $len = len_{max} + 1$
7. $C$ = PUSH($C_{max}$, $p$) /* add $p$ to $C_{max}$ to give new contour $C$ */
8. **return**

MULTIPLE SCALE ALGORITHM.
procedure FIND CONTOUR
1. get next POSSIBLE EDGE point, $p$
2. FOLLOWEDGE($p, C, wt, len$) in forward direction
3. FOLLOWEDGE($p, D, wt, len$) in backward direction
4. **if not** CLOSED($C$, $D$) **then** EXTEND($C$)
5. **if not** CLOSED($C$, $D$) **then** EXTEND($D$)
6. COMBINE($C$, $D$) into a single contour

procedure FOLLOWEDGE($p, C, wt, len$)
input: $p$, point to begin contour
output: $C$, contour points
$wt$, average weight of $C$
$len$, number of pixels in $C$
1. **if not** POSSIBLE EDGE($p$, *scale*) /* **true** if $p$ is possible edge at *scale* or coarser. */**then** $wt = len = 0$, **return**
2. *scale* = largest scale for which $p$ is a POSSIBLE EDGE
3. **for** $i = 1 \ldots 5$ **do** FOLLOWEDGE($p_i, C_i, wt_i, len_i$) /* see Fig. 1a for numbering of points */
4. $max = i$ where $wt_{max}$ is the largest of $wt_i$
5. **if** $wt_{max} = 0$ **then** /* $p$ has no continuation */
   $wt = factor1\_wt + factor2\_wt + factor3\_wt + 1$
   len = 1
   $C$ = PUSH($\Phi$, $p$) /* create new contour $C$ containing $p$ */
   **return**
6. /* continuation found */
   $wt = (len_{max} * (wt_{max} - len_{max}) + factor1\_wt + factor2\_wt + factor3\_wt)/(len_{max} + 1) + factor4\_wt$
7. $len = len_{max} + 1$
8. $C$ = PUSH($C_{max}$, $p$) /* add $p$ to $C_{max}$ to give new contour $C$ */
9. **return**

procedure EXTEND($C$)
1. **while (true)**
2. $p$ = end point of contour $C$
3. *scale* = largest scale at which $p$ is a POSSIBLE EDGE
4. **if** *scale* is smallest possible **then break**
5. *scale* = *scale* − 1 /* continue end at next smaller scale */
6. **for** $i = 1, 2, 3, 4, 5$ /* see Fig. 1a for labeling of points */
   **if** POSSIBLE EDGE($p_i$, *scale*) and directions of $p$ and $p_i$ differ by no more than 2
   **then** FOLLOWEDGE($p_i, C_i, wt_i, len_i$)
7. $max = i$, where $wt_{max}$ is largest of $wt_i$ /* find best extension */
8. **if** $wt_{max} > 0$ **then** add contour $C_{max}$ to contour $C$
   **else break** /* no extension */
9. **repeat**

## REFERENCES


1. V. Berzins, Accuracy of laplacian edge detectors, *Comput. Vision Graphics Image Process.* **27**, 1984, 195–210.
2. F. Bergholm, Edge focusing, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-9**, No. 6, 1987, 726–741.
3. W. F. Bischof and T. Caelli, Parsing scale-space and spatial stability analysis, *Comput. Vision Graphics Image Process.* **42**, 1988, 192–205.
4. J. F. Canny, *Finding Edges and Lines in Images*, Master's thesis, MIT, 1983.
5. J. J. Clark, Singularities of contrast functions in scale space, in *Proceedings, First International Conf. on Computer Vision, 1987*, pp. 988–1000.
6. J. J. Clark, Singularity theory and phantom edges in scale space, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-10**, 1988, 720–727.
7. J. Eklundh, T. E. Elfving, and S. Nyberg, Edge detection using the Marr–Hildreth operator with different sizes, in *ICPR-6, 1980*, pp. 1109–1112.
8. M. A. Fischler and H. C. Wolf, Linear delineation, in *Proc. IEEE Conf. on Computer Vision and Image Processing, 1983*, pp. 351–356.
9. I. Katz, *Coaxial Stereo and Scale-Based Matching*, Technical Report, Laboratory for Computational Vision, Department of Computer Science, University of British Columbia, 1985.
10. R. Leipnik, The extended entropy uncertainty principle. *Inform. and Control* **3**, 1960, 18–25.
11. D. Marr and E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London Ser. B*, **207**, 1980, 187–217.
12. A. Martelli, An application of heuristic search methods to edge and contour detection, *Commun. ACM* **19**, No. 2, 1976, 73–83.
13. R. Nevatia and K. R. Babu, Linear feature extraction and description, *Comput. Graphics Image Process.* **13**, 1980, 257–269.
14. A. M. Peich, Comments on fingerprints of two-dimensional edge models, *Comput. Vision Graphics Image Process.* **42**, 1988, 381–386.
15. A. Rosenfeld and M. Thurston, Edge and curve detection for visual scene, *IEEE Trans. Comput.* **C-20**, 1971, 562–569.
16. B. G. Schunck, *Gaussian Filters and Edge Detection*, Technical Report GMR-5586, General Motors Research Laboratories, 1986.
17. M. Shah, A. Sood, and R. Jain, Pulse and staircase edge models. *Comput. Vision Graphics Image Process.* **34**, 1986, 321–341.
18. V. Torre and T. Poggio, On edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, No. 2, 1986, 147–163.
19. D. Williams and M. Shah, *Edge Contours*, Technical Report CS-TR-88-18, University of Central Florida, Computer Science Department, Sept. 1988.
20. D. Williams and M. Shah, Edge contours using multiple scales, in *Applications of Artificial Intelligence VII* (M. M. Trivedi, Ed.), pp. 13–24, SPIE, Anaheim, CA, 1989.
21. A. Witkin, Scale-space filtering, in *International Joint Conf. on A.I., 1983*, pp. 1019–1021.