Edge detection by curve fitting

Ardeshir Goshtasby and Hai-Lun Shyu

Edge detection is formulated as a curve fitting problem. First, high-gradient pixels are grouped into elongated regions and then a curve is fitted to each. The curve fitting method used in this work does not require solving a system of equations, and therefore is fast. Examples of edge detection by curve fitting on synthetic and real images are presented, and results obtained are compared with those determined by the Laplacian of Gaussian operator.

Keywords: computer vision, image segmentation, edge detection, curve fitting, rational Gaussian curve, minimum-spanning tree

Edges signify sharp intensity variations in an image and provide information about the locations and shapes of objects[1]. Many studies have used image second derivatives to determine edges in an image[2-7]. In this paper, image first derivatives (gradients) will be used to determine edges.

The approach proposed here is in spirit similar to that described by Burns *et al.*[8] for the extraction of straight lines in an image. In Burns *et al.*'s method, regions were formed by grouping pixels according to their edge directions. Then a line was fitted to each obtained region. In the approach proposed here, regions are formed from pixels with gradient magnitudes above a threshold value. Then, regions are partitioned at their branch points, and finally a curve is fitted to each obtained region.

To represent edge contours by curve segments, Kass *et al.*[9] and Cohen[10] used energy-minimizing curves called *snakes*. In the snake model, there is no need to isolate elongated regions from an image, but there is a need to specify the initial positions of the snakes. Once a snake is initialized, it freely moves in its neighbourhood to minimize its energy while responding to external forces defined by local image edges.

Electrical Engineering and Computer Science Department, University of Illinois at Chicago, 1120 Science and Engineering Offices, 851 South Morgan Street, Chicago, IL 60607-7053, USA
Paper received: 3 June 1994; revised paper received: 23 August 1994

Curve fitting has been studied extensively[11-14]. Most techniques, however, use piece-wise polynomials, which require solving a system of equations to determine their coefficients. Since a region may contain a large number of points and have a complex structure, polynomial functions are in general not appropriate for our purposes. In this work, a formulation which defines a curve by a weighted sum of given points will be used. In this formulation, a curve is immediately obtained by substituting the coordinates of given points into the provided equations.

In the following, first the curve fitting technique which is used in this work is described. Then an algorithm which subdivides a complex region into simple ones, each of which can be replaced by a curve segment, is introduced. Next, the sensitivity of the proposed method to its free parameters is investigated, and finally, results of edge detection by curve fitting are presented and compared with those determined by the LoG operator. In this paper, by *given points* we mean high-gradient *pixels* in a region. From pixels in a region a minimum spanning tree (MST) is obtained. The *vertices* of the MST are then used as the *control points* of the approximating parametric curve. Therefore, *given points*, *pixels* in a region, MST *vertices* and *control points* of a curve all refer to the same physical entity.

RATIONAL GAUSSIAN CURVES

A rational Gaussian (RaG) curve is defined by[15]:

$$\mathbf{P}(u) = \sum_{i=1}^{n} \mathbf{V}_i g_i(u) \tag{1}$$

where $\mathbf{V}_i$ is the ith control point (the ith pixel in a region):

$$g_i(u) = \frac{W_i G_i(u)}{\sum_{j=1}^{n} W_j G_j(u)} \tag{2}$$

0262-8856/95/$9.50

is the ith basis function of the curve, W_i is the weight associated with the ith control point, and:

$$G_i(u) = e^{-(u-u_i)^2/2\sigma_i} \quad (3)$$

is a Gaussian of height one, u_i in formula (3), which is known as the ith node of the curve, is the parameter value at which the ith basis function is centred. The nodes of a curve have to be estimated from its control points. σ_i is the standard deviation of the ith Gaussian, and shows the amount of smoothing to be applied to the ith point and points in its neighbourhood. The standard deviation σ_i and parameter u_i are measured with the same unit.

Formulas (1)–(3) show an open curve. To obtain a closed curve, formula (3) should be replaced by[15]:

$$G_i(u) = \sum_{j=-\infty}^{\infty} e^{-[u-(u_i+j)]^2/2\sigma_i^2} \quad (4)$$

The infinity in this formula comes from the fact that a Gaussian extends from $-\infty$ to $+\infty$, and when a curve is closed it makes infinite cycles. A Gaussian, however, approaches zero exponentially, and in practice the infinity may be replaced by a small number. Assuming the accuracy of a computer is ε and the standard deviations of all Gaussians are equal to σ, we find:

$$-\lceil \sigma\sqrt{-2\ln\varepsilon}\,\rceil \leqslant j \leqslant \lceil \sigma\sqrt{-2\ln\varepsilon}\,\rceil \quad (5)$$

For conceivable values of ε and σ in digital images, it has been shown that it is sufficient to vary j in the range [−5,5].

A nice property of this representation is that it is not required to solve a system of equations to obtain a curve. A curve is immediately obtained by substituting the coordinates of given points into relations (1)–(3). The standard deviations of Gaussians in this formulation control the smoothness of the obtained curve. This is because the Fourier transform of a Gaussian in the spatial domain is another Gaussian in the frequency domain[16], and as the standard deviation of the Gaussian in the spatial domain increases, the standard deviation of the Gaussian in the frequency domain decreases, showing that the curve obtained will have smaller high-spatial-frequency coefficients. Since a curve with smaller high-spatial-frequency coefficients implies a smoother curve, we can conclude that as the standard deviations of Gaussians increase, a smoother curve will be generated. Hence, the ability to modify the standard deviations of Gaussians makes it possible to control the smoothness of generated edge contours.

The weight at a control point determines the degree of importance of that point with respect to others in defining a curve. In this work, the weight at a point is taken to be the gradient magnitude at the point. Taking the weights in this manner ensures that a curve will pass closer to points with higher gradient magnitudes.

Note that formulas (1)–(3) assume that given points are ordered. Since the order of pixels in a region is not known, an algorithm that orders pixels in a region is needed. We will develop this algorithm later, but first an algorithm is developed which partitions a complex region into simple ones, each of which can be approximated by a curve segment.

REGION SUBDIVISION

Consider a region which has been obtained from high-gradient pixels in an image. The region may represent a complex structure with many branches and loops. In this section an algorithm which divides such a region into simple ones, each of which can be replaced by a curve segment, is described.

A region will be subdivided using its minimum-spanning tree (MST)[17]. Given a region, its MST is determined and branches that are longer than a threshold value care cut off. Each obtained branch is treated like a new tree, and the subdivision is repeated until all trees have a trunk with branches that are shorter than a prespecified threshold value. A curve is then fitted to the vertices in each obtained tree.

Since distances between all adjacent pixels in a region are equal to a constant value, many MSTs can be obtained from a given region. For instance, two of the MSTs obtained from the region of *Figure 1a* are shown in *Figures 1b* and *1c*. *Figure 1b* will produce three parallel curve segments which cannot truly represent an object boundary, while *Figure 1c* will produce a single curve segment which more realistically represents an object boundary.

To obtain MSTs that are more like *Figure 1c* than *Figure 1b*, the gradients and intensities of pixels are taken into consideration when choosing the MST edges. An MST is grown gradually, starting from a single pixel until all given pixels are used. At any stage, the procedure that selects the next pixel among the unprocessed ones in a region takes the pixel which is closest to the MST. Since distances between all adjacent pixels in a region are the same, it is possible to obtain two or more pixels that can be selected as the next MST vertex. For instance in *Figure 1d*, any of pixels A, B or C can be selected as the next MST vertex. In such a situation, we will choose the pixel with the highest gradient magnitude. Therefore, if the gradient magnitude at pixel B is larger than those at pixels A and C, pixel B will be chosen as the next MST vertex. Choosing MST vertices in this manner will ensure that pixels with locally-maximum gradient magnitudes form a spine in a region.

If a pixel is of the same distance to two or more MST vertices, it could be connected to the MST in two or more different ways. In such a situation, we will connect this pixel to the MST vertex, which results in the largest absolute intensity difference between the two points. For example, in *Figure 1e*, C can be connected to B or D. Assuming that the intensities of B, C and D are 10, 15 and 12, respectively, C will be connected to B since the absolute intensity difference between them is 5 while that between C and D is 3. Selecting MST edges in this

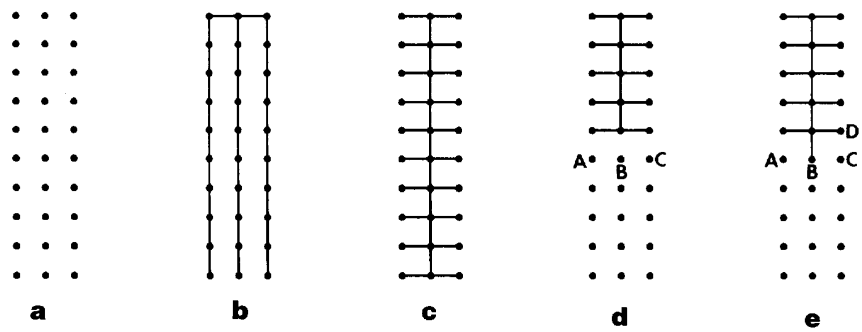

Figure 1 (a) Pixels in a region; (b), (c) two different MSTs of region (a); (d) any one of pixels A, B or C can be selected as the next MST vertex. Here, the pixel with the largest gradient magnitude is chosen; (e) pixel C can be connected to either of pixels B or D. C is connected to the pixel which has the largest absolute intensity difference with it

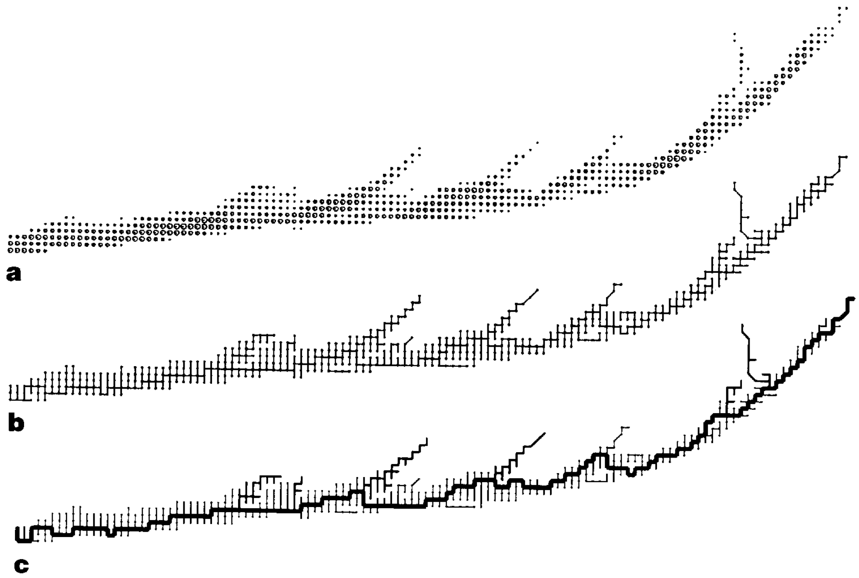

Figure 2 (a) Typical region in an image. The radii of the circles show the gradient magnitudes of the pixels; (b) the MST of the region; (c) pruning the tree branches. The thickest lines show the maximal path of the tree. The thinner lines show branches that are equal to or longer than seven pixels. The thinnest lines show the remaining branches that contain fewer than seven pixels

manner will ensure that branches are connected to the spine in the direction of the maximum gradient.

A complex region may have an MST that has many branches. To obtain longer curve segments, the maximal path (longest path) in the MST is determined. Then, branches connected to the maximal path that are longer than a threshold value are cut off. A branch is treated like a new tree, and if it has long branches they are recursively cut off until all trees have a trunk with branches that are shorter than a prespecified value.

Figure 2a shows a typical region obtained from high-gradient pixels in an image. Each pixel is shown by a circle with its radius proportional to the gradient magnitude of the pixel. The MST of this region is shown in *Figure 2b*, and the maximal path of the tree is shown with thick lines in *Figure 2c*. Branches that contain seven or more pixels are shown with thinner lines in *Figure 2c*. These branches will be cut off and treated like new trees. The pixels that remain will be used in curve fitting. For example in *Figure 2c*, pixels corresponding to the trunk (shown with the thickest lines) and branches shorter than seven pixels (shown with the thinnest lines) form a tree which will be used in curve fitting. This tree is shown in *Figure 3a*.

When fitting a parametric curve to a set of points, the order in which the curve should approximate the points must be known. In the next section, an algorithm which determines the order of pixels in a region (vertices in an MST) is described.

ORDERING THE MST VERTICES

When fitting a parametric curve to points $\{\mathbf{V}_i : i = 1, \ldots, n\}$, it is required that the order in which the curve approximates the points be given. In parametric curves, this ordering is provided by the nodes of the curve: $\{u_i : i = 1, \ldots, n\}$. The nodes of a curve show the positions of its basis functions in the parameter space. Node u_i shows the parameter value at which the effect of the ith control point is maximum on the curve. In this paper, the nodes of a curve are determined such that the effect of a control point is maximum at the curve point closest to it. If the curve were given, the curve point closest to a control point could be determined. However, since a curve cannot be generated before its nodes are known, the nodes of a curve cannot be computed in this manner. We use the maximal path of the MST of a region as an approxima-

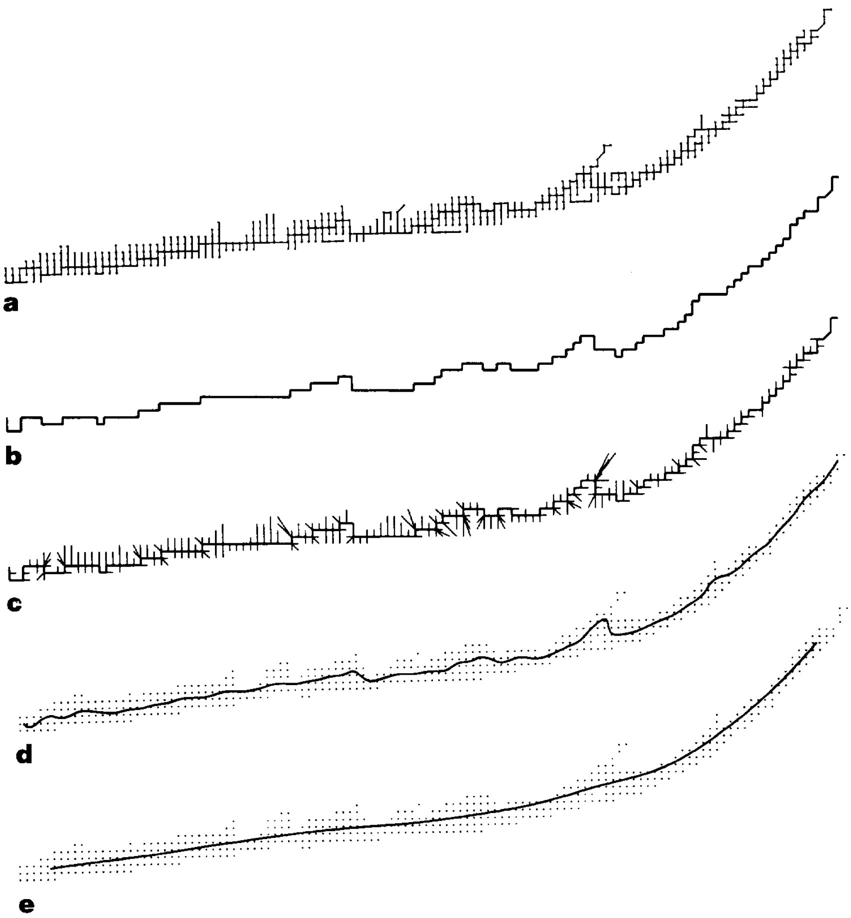

Figure 3 (a) An MST; (b) the maximal path of the MST; (c) computation of the nodes of a curve. For points on the maximal path, their distances to one end of the path are used as the nodes. For points not on the path, points on the path closest to them are determined, as shown here.

tion to the curve to be found and estimate its nodes. For points that lie on the maximal path, their distances to one end of the path are used as the corresponding nodes. For points that are not on the maximal path, their projections to the path (points on the path closest to them) are determined and distances of their projections to the same end point are used as the nodes. Estimating the nodes in this manner ensures that the effect of a control point becomes maximum at the curve point which is closest to it.

Figure 3 demonstrates the node selection process. *Figure 3a* shows the MST of *Figure 2c* after removal of the six small branches. The maximal path of the tree is shown in *Figure 3b* and the projections of points that are not on the path to the path are shown in *Figure 3c*. If the vertices of an MST are used as the $\mathbf{V}_i$'s in formula (1), the gradient magnitudes of the points as W_i's in formula (2), the nodes as the u_i's in formula (3), and the standard deviations of Gaussians $\sigma_i = 1$ pixel in formula (3), the curve shown in *Figure 3* will be obtained. The curve has picked up small image details. By increasing the standard deviations of Gaussians to five pixels we will obtain a smoother curve as shown in *Figure 3e*.

Because an MST cannot contain loops, the procedure described above will replace a closed boundary by an open curve. To avoid that, whenever two branch endpoints in an MST are adjacent, they are connected and a closed curve is fitted to the closed region. The spine in such a region will show a closed path. Branches attached to the spine which are longer than a threshold value are again cut off and treated like new trees.

The proposed formulations require that parameters $\{\sigma_i : i = 1, \ldots, n\}$ and $\{W_i : i = 1, \ldots, n\}$ be specified before a curve can be drawn. W_i will be set equal to the gradient magnitude at pixel $\mathbf{V}_i$. This will ensure that a pixel with a higher gradient magnitude will have a larger influence on a generated curve than a pixel with a lower gradient magnitude. In this paper, all standard deviations of Gaussians will be set to the same parameter σ so that the smoothness of all edge contours in an image could be modified by varying a single parameter. In the next section, the sensitivity of the proposed edge detection method to these parameters are investigated.

EDGE DETECTION PARAMETERS

Since images come from a variety of scanners, have different levels of noise and are in different resolutions, the edge detection methods employed should have parameters that can be adjusted to these image variations. The proposed method has three parameters which can be used for this purpose. They are: (1) the gradient threshold g which selects the pixels that are used in curve fitting; (2) the maximum branch-length l which ensures that a curve is fitted to a tree with branches shorter than a threshold value. This threshold value also removes noisy regions that contain fewer than l pixels; (3) the standard deviation of Gaussians σ which controls the smoothness of edge contours obtained.

The gradient threshold value should be selected taking into consideration the degree of detail in an image. In a noisy image, this threshold value should be increased to avoid detection of noisy edges. In a blurred image, this threshold value should be decreased to pick up region boundaries that may have low gradient magnitudes. As the gradient threshold is decreased, the number of points used in curve fitting increases, slowing down the edge detection process. As the gradient threshold is increased, the process becomes faster but some of the edge contours become disconnected. This effect can be observed in the example demonstrated in *Figure 4*. *Figure 4a* shows an image of a laboratory scene and *Figures 4b* and *4c* show regions containing pixels with gradient magnitudes equal to and greater than 16 and 20, respectively. Curves fitting to regions in *Figures 4b* and *4c* are depicted in *Figures 4d* and *4e*, respectively.

The maximum allowed branch length in a tree determines the positional accuracy of an obtained curve. An MST with long branches will produce a curve that may not reconstruct local image details well. On the other hand, if very small branches are cut off, very short curve segments will be obtained. A textured image containing a lot of details requires a small threshold value so that image details are not averaged out. A blurred image containing thick region boundaries requires a large threshold value so that multiple contours are not generated for the same region boundary. *Figures 4d* and *4f* compare curves fitting to regions of *Figure 4b* with branch-length thresholds of 10 and 15 pixels, respectively.

The standard deviation of Gaussians determines the smoothness of obtained curves. If σ is too large, edge contours with sharp corners cannot be reproduced. On the other hand, if σ is too small, unwanted noisy details may appear in generated contours. For a noisy image, a larger σ should be used to avoid detection of noisy edges. For a high-contrast image with very little noise, a smaller σ should be used to allow a curve to reproduce sharp corners. *Figures 4d* and *4g* compare curves fitting to the regions of *Figure 4b* with σ equal to one and three pixels, respectively.

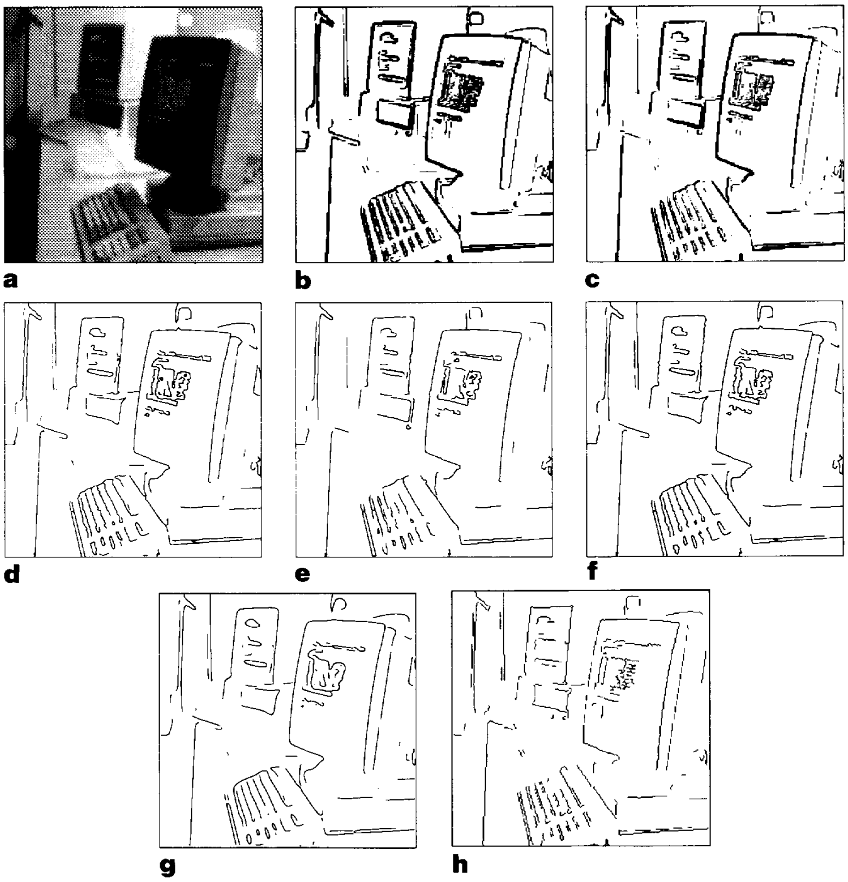

Figure 4 (a) 256 × 256 image of a laboratory scene; (b) pixels with gradient magnitudes equal to and greater than 16; (c) pixels with gradient magnitudes equal to and greater than 20; (d)–(g) edges determined by the proposed method with $g = 16$, $l = 10$, $\sigma = 1$; $g = 20$, $l = 10$, $\sigma = 1$; $g = 16$, $l = 15$, $\sigma = 1$; and $g = 16$, $l = 10$, $\sigma = 3$, respectively; (h) edges determined by the LoG operator with parameters similar to those used in (d), that is, $\sigma = 1$, edges with gradient magnitudes smaller than 16 were removed, and edge contours shorter than 10 pixels were also removed

Comparing *Figures 4d–4g*, we see that the curve fitting method is stable under variations of its parameters. That is, varying its parameters slightly will not change the detected edges drastically.

To evaluate the quality of edges obtained by the proposed method, obtained results were compared with those determined by an existing method. The LoG operator which is widely used in edge detection was chosen for comparison. Edges of the LoG operator determined with $\sigma = 1$ pixel and whose gradient magnitudes were equal to and greater than 16 are shown in *Figure 4h*. Comparing these edges with those determined with the same parameters by the curve fitting method (*Figure 4d*), we see that edges obtained by the curve fitting method are in general longer and smoother. To quantitatively compare the positional accuracy of edges determined by the curve fitting method and the LoG operator, in the next section a synthetic image with known edge positions is used.

RESULTS

To evaluate the accuracy of edges determined by the curve fitting method, a synthetic image, as shown in *Figure 5a*, was generated. This image contains four objects: a triangle, a square, a circle and a polygon. The intensities of the objects and the background were 200 and 150, respectively. The objects are of known sizes and are at known positions in the image. Therefore, the positions of true edges can be determined. Zero-mean Gaussian noise whose standard deviation σ_{noise} was such that signal-to-noise ratio (SNR) defined as[18] $step\text{-}size/\sigma_{noise}$ was equal to ten was added to *Figure 5a* to obtain *Figure 5d*. *Step-size* was the intensity difference between the objects and the background. Decreasing the signal-to-noise-ratio to five and two, *Figures 5g* and *5j*, respectively, were obtained.

Edges of *Figure 5a* determined by curve fitting and LoG operator are shown in *Figures 5b* and *5c*, respectively. The standard deviation of Gaussians in both methods was equal to one pixel, and the gradient threshold value for both methods was 20. That is, parameter g in the curve fitting method was set to 20, while edges obtained by the LoG operator whose gradient magnitudes were smaller than 20 were removed. Parameter l in the curve fitting method was set to ten, and all edge contours in the output of the LoG operator shorter than ten pixels were removed. Since positions of true edges were known, the distances between the true edges and edges obtained by curve fitting could be determined. The maximum and mean of such distances were determined and shown in *Table 1*. Maximum and mean errors for edges determined by the LoG operator were also computed and shown in *Table 1*. Note that since the curve fitting method generates a continuous contour, it is possible to determine the edge positions with subpixel accuracy.

Edges of *Figure 5d* determined by the curve fitting method and the LoG operator are shown in *Figures 5e* and *5f*, respectively. Here again, $\sigma = 1$, $l = 10$ and $g = 20$ were used. The maximum and mean edge positional errors are shown in *Table 1*. For *Figure 5g*, since SNR was rather low, edges extracted with $\sigma = 1$ pixel were very noisy. Therefore, σ was increased to two pixels. The gradient threshold was also reduced to ten so that smoothed object boundaries were not missed in curve fitting. l was kept at ten pixels. Edges obtained by curve fitting and LoG operator are shown in *Figures 5h* and *5i*, respectively. Their maximum and mean positional errors were also determined, and entered into *Table 1*. Finally, edges of *Figure 5j* determined by the curve fitting and the LoG operator are shown in *Figures 5k* and *5l*, respectively. Since SNR was considerably low in image *5j*, σ was further increased to three pixels and the gradient threshold was further decreased to five. The maximum and mean errors of edges determined by the two methods are shown in *Table 1*.

As can be observed from *Figure 5*, when images are not noisy, both methods perform well. However, as image noise increases, the curve fitting method produces longer and smoother edges, demonstrating that the curve fitting method is less sensitive to noise than the LoG operator. We can arrive at the same conclusion by examining the entries of *Table 1*. Because of the continuous nature of edges determined by curve fitting, it is possible to determine the edge positions with subpixel accuracy. *Table 1* shows that when the curves are quantized to discrete image pixels, errors obtained by the curve fitting method are about half of those determined by the LoG operator.

To provide further experimental results, four real images as shown in *Figures 6a*, *7a*, *8a* and *9a* were used. *Figure 6a* shows a chest radiograph. Boundary contours of this image computed by curve fitting using $\sigma = 1$ pixel, $g = 10$ and $l = 10$ pixels are shown in *Figure 6b*. These parameters are the same as those used in *Figure 4d*, except that g was reduced to ten because contrast of *Figure 6a* was lower than that of *Figure 4a*. To evaluate the quality of obtained contours, *Figures 6a* and *6b* were overlaid to obtain *Figure 6c*. Boundary contours determined by the LoG operator with parameters similar to those used in *Figure 6b* were also determined, and shown in *Figure 6d*. That is, $\sigma = 1$, and edges obtained by the LoG operator with gradient magnitudes smaller than 16 were removed. In addition, edge contours shorter than ten pixels were removed. Very similar edge contours have been obtained by both methods. The curve fitting method has produced slightly longer and smoother contours near the sides of the image.

Figure 7a shows an X-ray angiogram. Using the same parameters as in *Figure 6b*, the curve fitting method generated the edge contours shown in *Figures 7b* and *7c*. Edge contours determined by the LoG operator with parameters similar to those used in *Figure 7b* are shown

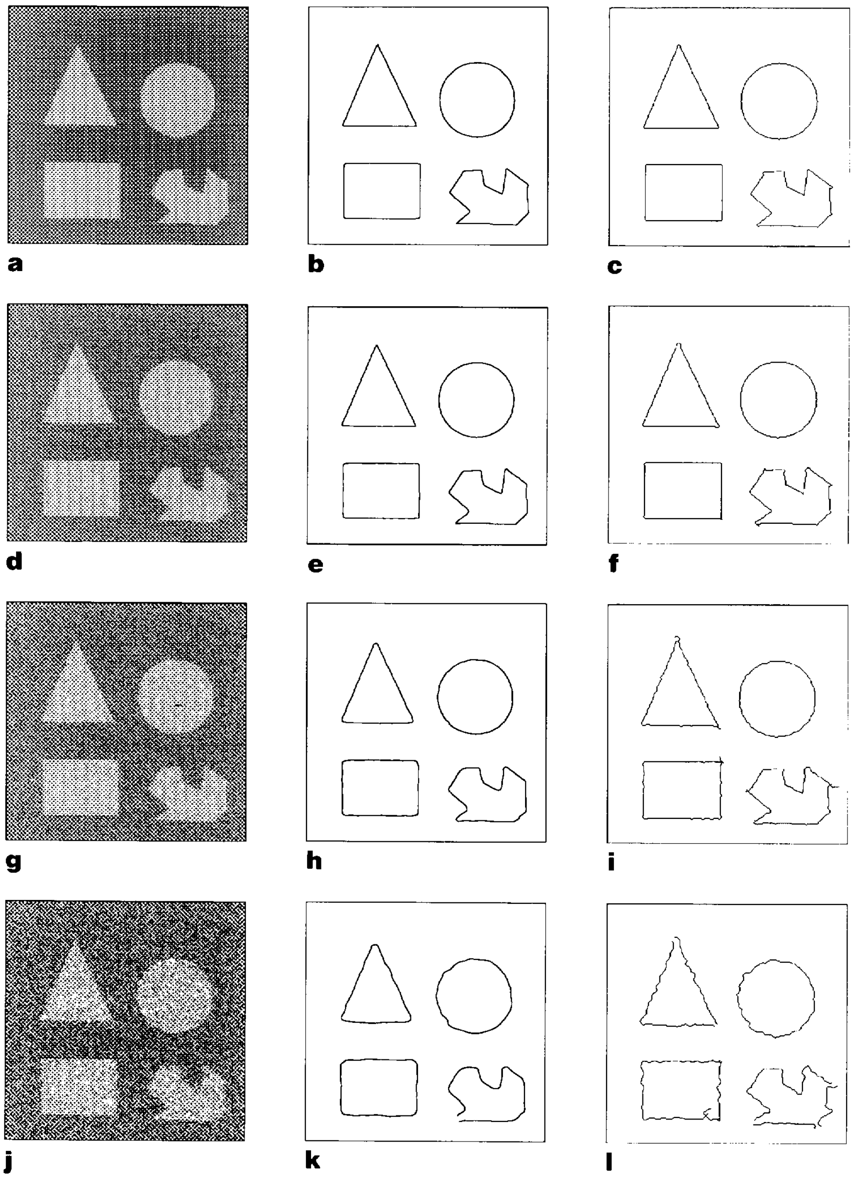

Figure 5 (a) 256 × 256 synthetic image with known edges. Zero-mean Gaussian noise was added to (a) to obtain images (d), (g) and (j) with SNR of 10, 5 and 2, respectively. The middle-column images show edges determined by the curve fitting method and the right-column images show edges obtained by the LoG operator. The same parameters were used by both curve fitting and LoG operator

in *Figure 7d*. Comparing *Figures 7c* and *7d*, we see that the two methods have produced very similar edge contours, with the LoG operator contours being slightly longer than those determined by the curve fitting method.

Figure 8a shows a Landsat Thematic Mapper image of a terrain scene. Since image contrast in *Figure 8a* is higher than that of *Figure 4a*, parameter g was increased to 20, but other parameters were kept the same as those used in *Figure 4d*. Edge contours obtained by curve fitting are shown in *Figures 8b* and *8c*. Edge contours determined by the LoG operator with the same parameters are shown in *Figure 8d*. Significant region boundaries such as the lake boundaries have been detected equally well by both methods. Small contours corresponding to the terrain features are extracted differently by the two methods. It is not possible to tell the superiority of one method over another in these images.

As the final example, the aerial photograph shown in *Figure 9a* was used. Parameters used to determine the edges of *Figure 8b* were used here also. Edges obtained by the curve fitting method and the LoG operator are shown in *Figures 9c* and *9d*. The LoG operator has picked up the major roads better than the curve fitting method. The curve fitting method has merged some segments of contours from the two sides of the roads into single contours due to the small width of the roads. When the roads become very narrow, however, the curve fitting method has been able to detect longer road segments, especially near the centre of the image.

Table 1 Accuracy of edge positions determined by the curve fitting method and the LoG operator. Table entries show maximum and mean distances between the true edges of *Figure 5a* and edges determined by the curve fitting method and the LoG operator as a function of the signal-to-noise ratio (SNR)

SNR	*Curve fitting*		*LoG operator*	
	Mean	*Max*	*Mean*	*Max*
∞	0.1	0.5	0.2	1.0
10	0.3	1.2	1.2	3.0
5	0.7	3.3	1.8	6.0
2	1.2	5.5	2.7	12.0

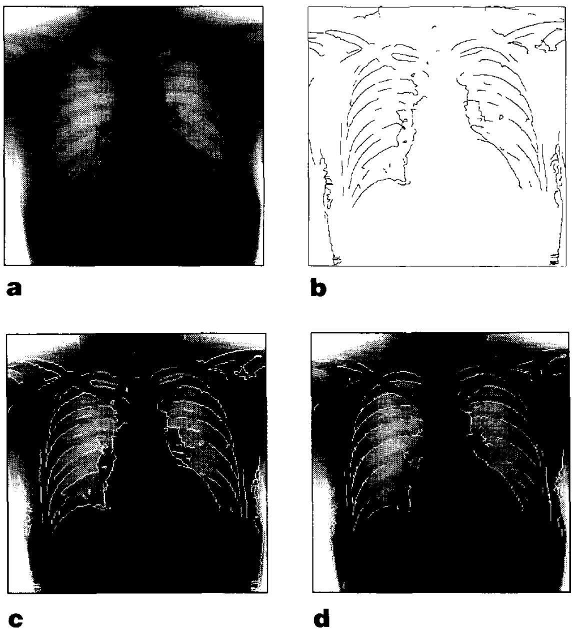

Figure 6 (a) 256 × 256 chest radiograph; (b) edges of the radiograph obtained by curve fitting with $g = 10$, $l = 10$ and $\sigma = 1$; (c) overlaying of images (a) and (b) after digitization; (d) edges of the LoG operator determined with the same parameters as those used in (b), that is, $\sigma = 1$, edges with gradient magnitudes equal to and greater than 10 were removed, and edge contours shorter than 10 pixels were also removed

Computationally, the curve fitting method requires in the order of $n \log n$ square roots to determine the MST of a region containing n pixels[17], in the order of n comparisons to determine the maximal path of the region, and in the order of n^2 square roots to determine the nodes of the curve that approximates the region. Usually, a fraction of pixels in an image are used in curve fitting. For instance, in *Figures 5a, 5d, 5g* and *5j*, only 4%, 5%, 8% and 13% of the entire image pixels, respectively, were used to obtain the edge contours. Computation of the LoG operator even when FFT is used requires in the order of $N^2 \lg N$ multiplications[16], where image size is assumed to be $N \times N$. The computational complexity of the LoG operator is a function of the image size only, whereas the computational complexity of the curve fitting method depends on the image contents. *Figures 5b, 5e, 5h* and *5k* determined by the curve fitting method required 50, 62, 110 and 155 seconds processing time on a SUN SPARC IPC computer, while *Figures 5c, 5f, 5i* and *5l* determined by the LoG operator required a constant time of 95 seconds.

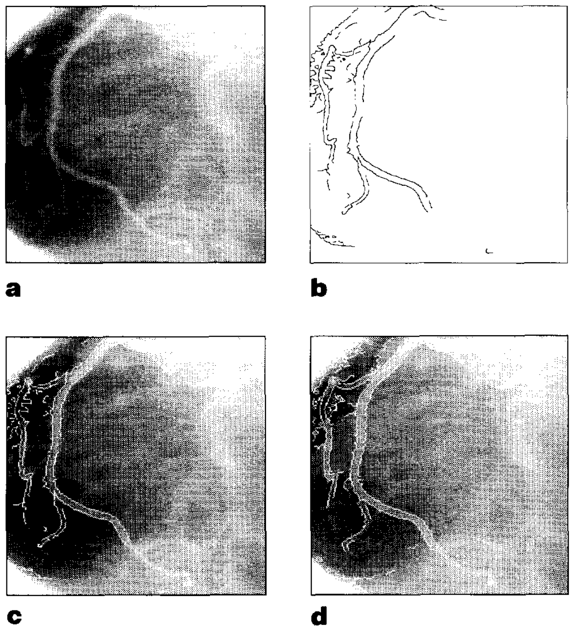

Figure 7 (a) 256 × 256 X-ray angiogram; (b) edges of the angiogram determined by curve fitting with parameters similar to those used in *Figure 6b*; (c) overlaying of images (a) and (b) after digitization; (d) edges determined by the LoG operator with parameters similar to those used in (b)

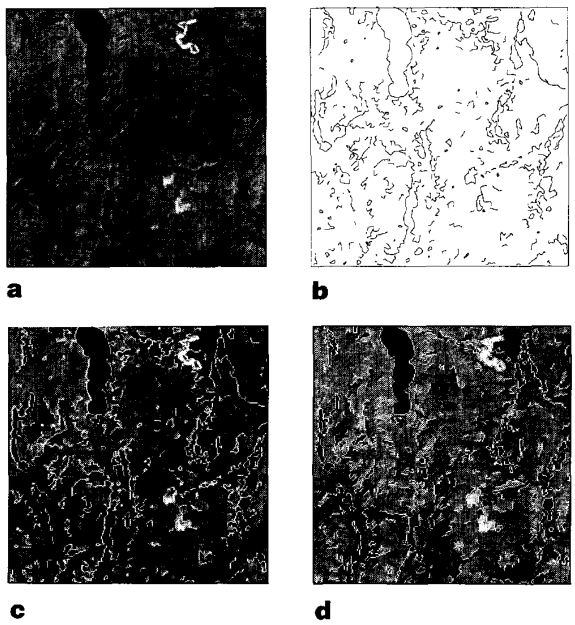

Figure 8 (a) 256 × 256 Landsat image; (b), (c) edges determined by curve fitting using $g = 20$, $l = 10$ and $\sigma = 1$; (d) edges determined by the LoG operator with parameters similar to those used in (b)

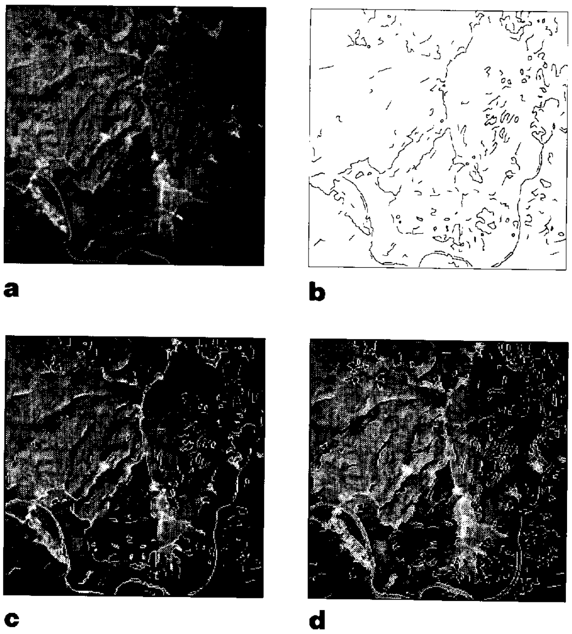

Figure 9 (a) 256 × 256 aerial photograph; (b), (c) edges determined by curve fitting using parameters similar to those used in *Figure 8b*; (d) edges determined by the LoG operator with parameters similar to those used in (b)

CONCLUSIONS

Edge detection is the first step in many image analysis problems. In this paper, a new edge detection method based on the idea of curve fitting was introduced. Weighted averaging was used to reduce noise and determine the positions of edges. The minimum-spanning tree was used as a tool to partition connected high-gradient pixels into elongated regions and fit a curve to each. The minimum-spanning tree was also used to determine the adjacency relation between pixels in a region, and to compute the nodes of a parametric curve that fits the region.

The proposed method has three parameters that can be adjusted to different image variations: (1) the standard deviation of Gaussian σ in the proposed method is analogous to the standard deviation of Gaussian used in the LoG operator. As image noise increases this parameter should be increased; (2) the gradient threshold value g is used to select high-gradient pixels for curve fitting. This parameter is analogous to the gradient threshold value used to remove weak and noisy edges obtained by the LoG operator[3], (3) the branch length threshold value l is used to cut off MST branches, and also to ensure that trees containing less than l pixels are not used in curve fitting. This parameter avoids the generation of noisy and short contours, and ensures that high edge positional accuracy is achieved by restricting the width of regions used in curve fitting to $2l+1$ pixels.

Qualitatively, results on four real images showed that edges obtained by curve fitting were similar to those determined by the LoG operator. Since edge contours obtained by the curve fitting method are in continuous form, it is possible to determine their positions with subpixel accuracy, and also generate them in arbitrary resolutions.

REFERENCES

1 Brooks, R A 'Model-based three-dimensional interpretations of two-dimensional images', *IEEE Trans. PAMI*, Vol 5 No 2 (1983) pp 140–150
2 Marr, D and Hildreth, E C 'Theory of edge detection', *Proc. Royal Soc. Lond.*, Vol 207 (1980) pp 187 217
3 Fleck, M M 'Some defects in finite difference edge finders', *IEEE Trans. PAMI*, Vol 14 No 3 (1992) pp 337–346
4 Haralick, R M 'Digital step edges from zero-crossing of second directional derivatives', *IEEE Trans. PAMI*, Vol 6 No 1 (1984) pp 58–68
5 Canny, J 'A computational approach to edge detection', *IEEE Trans. PAMI*, Vol 8 No 6 (1986) pp 679–698
6 Torre, V and Poggio, T A 'On edge detection', *IEEE Trans. PAMI*, Vol 8 No 2 (1986) pp 147–163
7 Deriche, R 'Fast algorithms for low-level vision', *IEEE Trans. PAMI*, Vol 12 No 1 (1990) pp 78–87
8 Burns, J B, Hanson, A R and Riseman, E M 'Extracting straight lines', *IEEE Trans. PAMI*, Vol 8 No 4 (1986) pp 425–477
9 Kass, M, Witkin, A and Terzopoulos, D 'Snakes: Active contour models', *Int. J. Comput. Vision*, Vol 1 (1988) pp 321–331
10 Cohen, L D 'On active contour models and balloons', *Image Understanding*, Vol 53 No 2 (1991) pp 211–218
11 Ichida, K, Kiyono, T and Yoshimoto, F 'Curve fitting by a one-pass method with piecewise cubic polynomial', *ACM Trans. Math. Software*, Vol 3 No 2 (1977) pp 164 174
12 Farin, G *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York (1989)
13 Faux, I D and Pratt, J M *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester (1979)
14 Lancaster, P and Salkauskas, K *Curve and Surface Fitting*, Academic Press, New York (1986)
15 Goshtasby, A 'Design and recovery of 2-D and 3-D shapes using rational Gaussian curves and surfaces', *Int. J. Comput. Vision*, Vol 10 No 3 (1993) pp 233 256
16 Castleman, K R *Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ (1981) pp 161–163
17 Preparata, F P and Shamos, M I *Computational Geometry: An Introduction*, Springer-Verlag, New York (1985) pp 219–233
18 Nawla, V S *A Guided Tour of Computer Vision*, Addison-Wesley, New York (1993) p 86