



0031–3203(95)00104–2

OPTIMUM APPROXIMATION OF DIGITAL PLANAR CURVES USING CIRCULAR ARCS

SOO-CHANG PEI and JI-HWEI HORNG

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

(Received 9 December 1994; in revised form 20 June 1995; received for publication 7 July 1995)

Abstract—Given a digital planar curve of N ordered points, the dynamic programming algorithm is applied to find M dominant points, among the N points, which construct a globally optimal approximation to the given curve provided that a circular arc is properly designed between each pair of adjacent dominant points. This curve-fitting method is generalized to approximate closed curves. A fast algorithm for efficient computation is also introduced. The performance is shown by some experimental results.

Curve-fitting Circular arc Dynamic programming Cyclic string

1. INTRODUCTION

Polygonal approximation is a very important topic in the area of digital image processing. Many approaches have been proposed. An attractive approach among these is to formulate the approximation as an optimization problem.^(1–3) The algorithms,^(1,2) proposed by Ray recursively maximize the objective function to find the j th vertex subject to the constraint that the $(j-1)$ th vertex is fixed. Perez⁽³⁾ applied the dynamic programming algorithm to minimize the error function. The polygonal approximation constructed by this algorithm is a globally optimal approximation.

Unfortunately, not all of the digital curves are suitable for applying the polygonal approximation. Smooth curves are usually hard to be approximated by using polygons. To improve the performance of approximation, high-order curves should be used. Splines are the most commonly used.^(4,5) However, the computational complexity is largely increased using splines. In this paper we use circular arcs to approximate digital curves. Smooth curves can be efficiently fitted and the property of low computational complexity is preserved. Only a few circular arcs are required to approximate a smooth curve. For a polygon-like curve, a set of circular arcs with small curvatures are automatically chosen by algorithm to fit the curve.

The dynamic programming algorithm is applied to find the optimal set of vertices. A simple approach is presented to estimate the parameters of the circular arcs, which are used to fit the given curve between each pair of adjacent vertices. The curve-fitting method is generalized to approximate closed curves. Finally, a fast algorithm is introduced to reduce the computational complexity. Several experimental results are also presented.

2. PROBLEM FORMULATION

The problem can be stated as follows: Given a digital curve of N ordered points, find M vertex points among those given and design a circular arc between each pair of adjacent vertices so that the resulting curve most closely fits the given digital curve. The set of N given points is denoted by:

$$S = \{s_1, s_2, \dots, s_N\}, \\ = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}.$$

The set of M vertices to be found is denoted by:

$$V = \{v_1, v_2, \dots, v_M\}.$$

The set of $M-1$ circular arcs to be designed is denoted by:

$$A = \{\text{arc}(v_1, v_2), \text{arc}(v_2, v_3), \dots, \text{arc}(v_{M-1}, v_M)\}.$$

where $\text{arc}(v_k, v_{k+1})$ is the circular arc designed to fit the given points between v_k and v_{k+1} .

The approximation error of the k th arc $\text{arc}(v_k, v_{k+1})$ is defined by:

$$e(v_k, v_{k+1}) = \sum_{s_i=v_k}^{v_{k+1}} d[s_i, \text{arc}(v_k, v_{k+1})], \quad (1)$$

where $d[s_i, \text{arc}(v_k, v_{k+1})]$ is the error measure between the point s_i and the k th arc $\text{arc}(v_k, v_{k+1})$.

The problem can now be formulated as follows: Given an ordered point set S of length N , find M vertices V among S , where $v_1 = s_1$ and $v_M = s_N$, and properly design a circular arc between each pair of adjacent vertices, such that the approximation error is minimized. That is, we seek to find the minimum error:

$$E(N, M) = \min_{V \subset S, A} \sum_{k=1}^{M-1} e(v_k, v_{k+1}), \\ = \min_{V \subset S, A} \sum_{k=1}^{M-1} \sum_{s_i=v_k}^{v_{k+1}} d[s_i, \text{arc}(v_k, v_{k+1})], \quad (2)$$

where the error measure d is defined by:

$$d[s_j, \text{arc}(v_k, v_{k+1})] = \{R_k^2 - [(x_i - x_c^k)^2 + (y_i - y_c^k)^2]\}^2. \tag{3}$$

The reason to choose this unusual error measure, which is neither integral absolute error nor integral square error and its effect on the problem will be discussed later.

3. DYNAMIC PROGRAMMING

The problem can be solved using dynamic programming, which is based on the recursive function:

$$E(n, m) = \min_{2m-3 \leq j \leq n-2} [E(j, m-1) + e(s_j, s_n)],$$

where $E(n, m)$ denotes the minimum approximation error of fitting $\{s_1, s_2, \dots, s_n\}$ using m vertices, $E(j, m-1)$ denotes the minimum approximation error of fitting $\{s_1, s_2, \dots, s_j\}$ using $m-1$ vertices and $e(s_j, s_n)$ denotes the minimum approximation error of fitting $\{s_j, s_{j+1}, \dots, s_n\}$ using a single arc, which ends at s_j and s_n .

The philosophy of dynamic programming is to derive $E(\cdot, m)$ from $E(\cdot, m-1)$ recursively. $E(n, m)$ is found by adjusting j , such that the sum of $E(j, m-1)$ and $e(s_j, s_n)$ reaches its minimum value. That is, the fitting of the first n points using m vertices is partitioned into the fitting of the first j points using $m-1$ vertices and the fitting of the remaining points by a single arc, which ends at the j th and n th points.

A circular arc can be designed to pass through three consecutive points; therefore, $2m-3$ points can be fitted without error by choosing the 1st, 3rd, ... and $(2m-3)$ th point ($m-1$ points in total) as vertices. That is, $E(2m-3, m-1)$ is equal to zero and $2m-3$ is the lower bound of j . Further reduction of j will not reduce $E(j, m-1)$; on the contrary, it will increase $e(s_j, s_n)$. Similarly, $n-2$ is the upper bound of j , because the $(n-2)$ th, $(n-1)$ th and n th point can be fitted without error by the additional arc.

To ensure that the first point will be chosen as the vertex, the value of $E(1, 1)$ is set at zero and the values of all the $E(n, 1)$ with $n \neq 1$ are set at infinity. Thereafter, the algorithm can be started from the initial state $E(\cdot, 1)$.

The dynamic programming algorithm is summarized as follows:

Algorithm 1.

```

E(1, 1) = 0.
For n = 2-N,
  E(n, 1) = +∞.
Endfor.
For m = 2-M,
  For n = 2m-1-N,
    E(n, m) = min_{2m-3 ≤ j ≤ n-2} [E(j, m-1) + e(s_j, s_n)].
  Endfor.
Endfor.
Return E(N, M).
    
```

The computation of $e(s_j, s_n)$ will be discussed in the next section.

4. ESTIMATION OF ARC PARAMETERS

To compute the approximation error $e(s_j, s_n)$, the circular arc $\text{arc}(s_j, s_n)$ should be designed in advance. The optimum values of the arc parameters depend on the definition of the error measure. The most commonly used error measures are integral absolute error, integral square error, and so on. However, the optimal values corresponding to these error measures are hard to be solved efficiently.

U.M. Landau has suggested an iterative algorithm for solving a similar problem.⁽⁶⁾ Since the algorithm is based on an iterative mechanism, its computational complexity is high. An efficient algorithm for solving Landau's problem is proposed by S.M. Thomas.⁽⁷⁾ The exact solution is found by carefully redefining the error function. He chose "area" instead of "length" as the quantity of the error measure. Due to the redefinition of error measure, an estimation bias results. This bias is small and approaches zero as the number of points approaches infinity.

In order to design the arcs efficiently, we use the modified error function defined by Thomas, which is given by equation (3). Therefore, the parameters, which are the center (x_c, y_c) and the radius R , of the arc (s_j, s_n) can be solved by minimizing the error function:

$$J = \sum_{i=j}^n \{R^2 - [(x_i - x_c)^2 + (y_i - y_c)^2]\}^2. \tag{4}$$

In our problem, the arc should pass through the points s_j and s_n . Therefore, the arc center must lie on the line:

$$y = ax + b,$$

where

$$a = -\frac{x_n - x_j}{y_n - y_j},$$

$$b = \frac{y_j + y_n}{2} - a \frac{x_j + x_n}{2}.$$

That is, y_c can be expressed in terms of x_c by:

$$y_c = ax_c + b. \tag{5}$$

The arc radius R can be expressed in terms of the arc center (x_c, y_c) by:

$$R^2 = (x_j - x_c)^2 + (y_j - y_c)^2. \tag{6}$$

Substituting equations (5) and (6) into (4), it becomes:

$$J = \sum_{i=j}^n \{[(x_j - x_c)^2 + (y_j - (ax_c + b))^2] - [(x_i - x_c)^2 + (y_i - (ax_c + b))^2]\}^2.$$

Differentiating with respect to x_c and setting to zero yields:

$$x_c = -\frac{\sum_{i=j}^n K_1 K_2}{\sum_{i=j}^n K_1 K_3}, \tag{7}$$

where

$$\begin{aligned} K_1 &= -x_j - ay_j + x_i + ay_i, \\ K_2 &= x_j^2 + (y_j - b)^2 - x_i^2 - (y_i - b)^2, \\ K_3 &= -2x_j - 2a(y_j - b) + 2x_i + 2a(y_i - b). \end{aligned}$$

Then, y_c and R are solved by substituting x_c back into equations (5) and (6). The approximation error is calculated by substituting all these parameters back into equation (4).

Remarks

- If $y_j = y_n$, equation (5) will not be valid. In such cases, the equation $x_c = (x_j + x_n)/2$ is used. Then the arc parameters can be solved by similar derivations.
- If the denominator of equation (7) is equal to or approximately equal to zero, a straight line segment is used to approximate the given points.
- The incremental computation method introduced in reference (3) can also be applied here to reduce the computational complexity.
- If we want to minimize integral square error, in other words, the error measure:

$$d[s_i, \text{arc}(v_k, v_{k+1})] = [R_k - \sqrt{(x_i - x_c^k)^2 + (y_i - y_c^k)^2}]^2 \quad (8)$$

is chosen instead of that given by equation (3), then equations (5), (6) and (7) provide good estimations of the optimal arc parameters. The estimation bias is small and approaches zero as the number of points approaches infinity. The approximation error can be calculated by substituting these parameters into the integral square error function:

$$e(s_j, s_n) = \sum_{i=j}^n [R - \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}]^2. \quad (9)$$

That is, our algorithm can be applied to construct suboptimal approximation of the given curve in least integral square error sense.

5. FITTING CLOSED CURVES

If the given digital curve is a closed curve, a straightforward method to find the globally optimal approximation is to try with all the N possible initial points and choose the minimum error outcome. In doing so, many of the approximation errors $e(s_j, s_n)$ are repeatedly used. The computational complexity can be largely reduced by carefully designing the algorithm.

Before proposing our algorithm, let us survey a similar work by Maes.⁽⁸⁾ In 1974, Wagner and Fischer proposed an algorithm for solving the string-to-string correction problem.⁽⁹⁾ An edit graph was constructed by using the edit operations. Then, the dynamic programming was applied to compute the edit distance. A generalized algorithm was proposed by Maes to handle cyclic strings.

Let us now apply the concept proposed by Maes to generalize our algorithm. The approximation errors used in Algorithm 1 for fitting curves with fixed initial

point are:

$$e(s_j, s_n), \quad \text{where } 1 \leq j \leq n \leq N.$$

To fit closed curves, the required approximation errors are:

$$e(s_j, s_n), \quad \text{where } 1 \leq j, n \leq N.$$

Notice that, the closed curve is treated as a cyclic string; that is:

$$\begin{aligned} S &= \{s_1, s_2, \dots, s_N\}, \\ &= \{s_k, s_{k+1}, \dots, s_N, s_1, \dots, s_{k-1}\}, \quad \forall k \in \{1, 2, \dots, N\}. \end{aligned}$$

To compute the approximation error $e(s_j, s_n)$ with $j \geq n$, the error of the string $\{s_j, s_{j+1}, \dots, s_N, s_1, \dots, s_n\}$ with respect to the arc $\text{arc}(s_j, s_n)$ is taken into consideration. The approximation errors $e(s_j, s_n)$ with $n - j = 0, 1$ and 2 are setted to zero without any effect on the result.

Since these approximation errors are repeatedly used, they are computed and saved in advance. Each time when an approximation error is required, it is picked out from the memory instead of computed once more.

In the k th loop of our algorithm, the k th point of the given digital curve is chosen as the initial point. The temporary digital curve to be fitted is:

$$S_k = \{s_k, s_{k+1}, \dots, s_N, s_1, \dots, s_{k-1}, s_k\}.$$

The length of the temporary curve is $N + 1$, because the curve to be designed must start from and terminate at the same point. The number of vertices to be used is $M + 1$. The last point of the temporary curve is a virtual point and will be chosen as the last vertex. This vertex will be merged with the first vertex, resulting in a closed curve of M vertices in total.

The algorithm for fitting closed curves is now summarized as follows:

Algorithm 2.

Compute the approximation error array ξ of size $N \times N$, where:

$$\xi(i, j) = e(s_i, s_j), \quad 1 \leq i, j \leq N.$$

Construct the appended array:

$$\xi_a = \begin{bmatrix} \xi & \xi \\ \xi & \xi \end{bmatrix}.$$

$$\tilde{N} = N + 1.$$

$$\tilde{M} = M + 1.$$

For $k = 1 - N$.

Construct the temporary approximation error array $\tilde{\xi}$, where:

$$\tilde{\xi}(i, j) = \xi_a(i + k - 1, j + k - 1), \quad 1 \leq i, j \leq \tilde{N}.$$

Call *Algorithm 1*.

$$E_a(k) = E(\tilde{N}, \tilde{M}).$$

Endfor.

$$E_m = \min_{k \in \{1, 2, \dots, N\}} E_a(k).$$

Return E_m .

6. EXPERIMENTAL RESULTS

In order to verify the performance of the present algorithm, we have applied our algorithm to the four digital curves used in reference (10), namely, a chromosome-shaped curve [Fig. 1(a)], a leaf-shaped curve [Fig. 2(a)], a figure-eight curve [Fig. 3(a)] and a curve with four semicircles [Fig. 4(a)]. The vertices chosen by the dynamic programming algorithm are marked by small circles and the solid lines are the constructed curves. As shown in the figures, our algorithm constructs good approximations of smooth curves as well as polygon-like curves.

In order to compare with the Teh-Chin algorithm,⁽¹⁰⁾ the integral square error given by equation

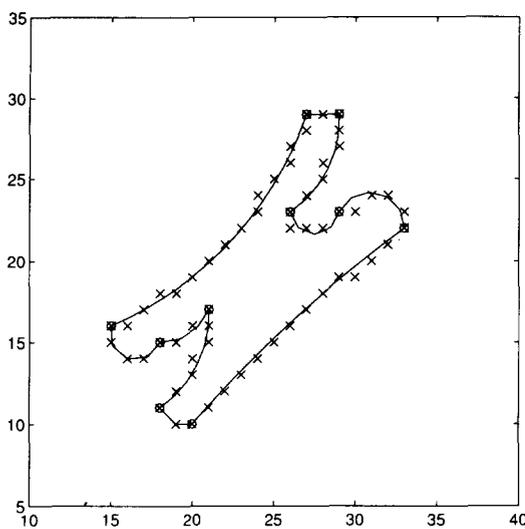


Fig. 1(a). Optimal approximation of a chromosome-shaped curve using circular arcs ($N = 60$, $M = 10$).

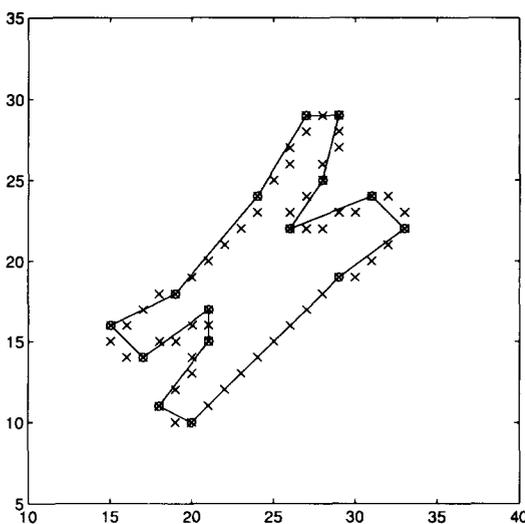


Fig. 1(b). Polygonal approximation of a chromosome-shaped curve using the Teh-Chin algorithm ($N = 60$, $M = 15$).

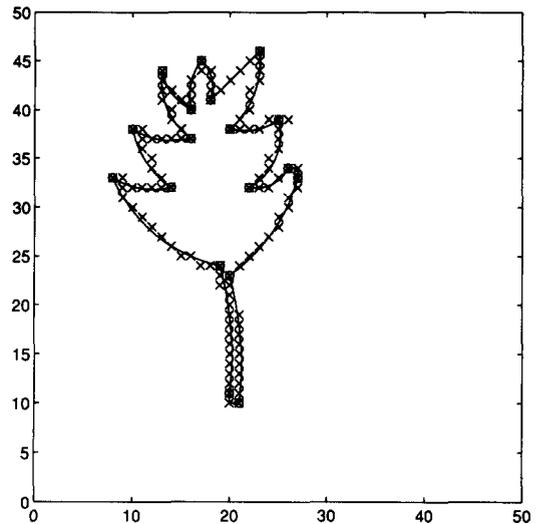


Fig. 2(a). Optimal approximation of a leaf-shaped curve using circular arcs ($N = 120$, $M = 18$).

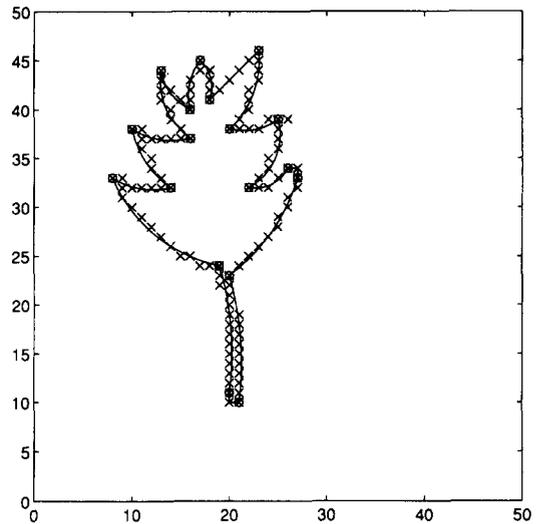


Fig. 2(b). Polygonal approximation of a leaf-shaped curve using the Teh-Chin algorithm ($N = 120$, $M = 29$).

(8) is chosen as the error measure in our experiments. As discussed in the last remark of Section 4, suboptimal approximations of the given curves are constructed by our algorithm. From another point of view, the resultant curves are optimal approximations of the given curves subject to the constraint that the parameters of all circular arcs are estimated using equations (5), (6) and (7). Of course, these equations provide good estimations of the optimal values.

The Teh-Chin algorithm⁽¹⁰⁾ is a famous algorithm for constructing polygonal approximations of digital curves. Teh and Chin are the first to propose the concept that the region of support of each point should be determined independently. Their algorithm provides excellent polygonal approximations of digital

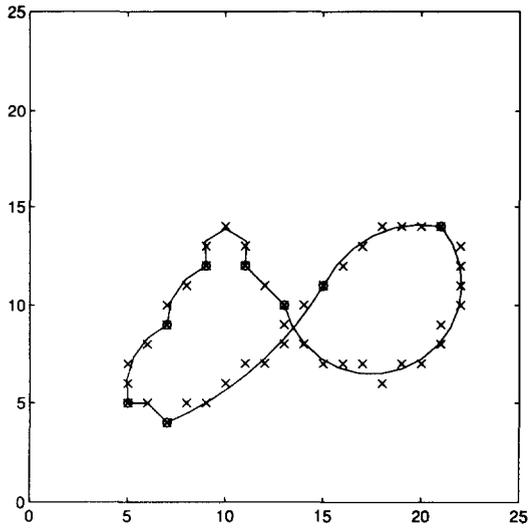


Fig. 3(a). Optimal approximation of a figure-eight curve using circular arcs ($N = 45, M = 8$).

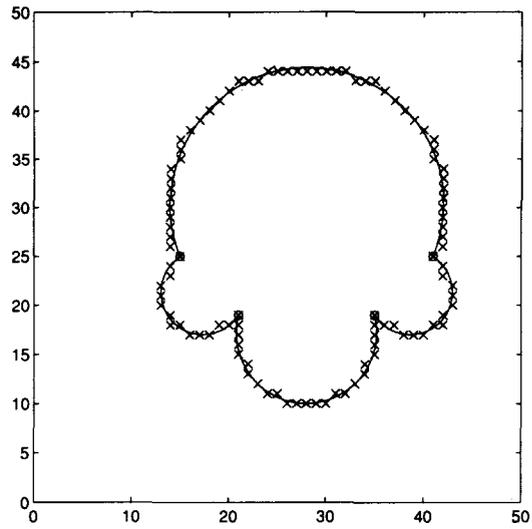


Fig. 4(a). Optimal approximation of a curve with four semicircles using circular arcs ($N = 102, M = 4$).

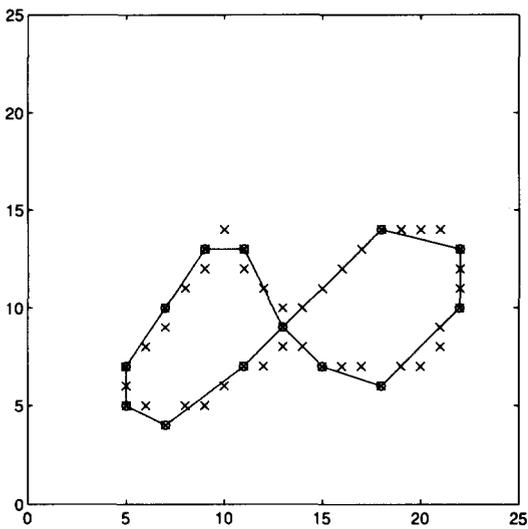


Fig. 3(b). Polygonal approximation of a figure-eight curve using the Teh-Chin algorithm ($N = 45, M = 13$).

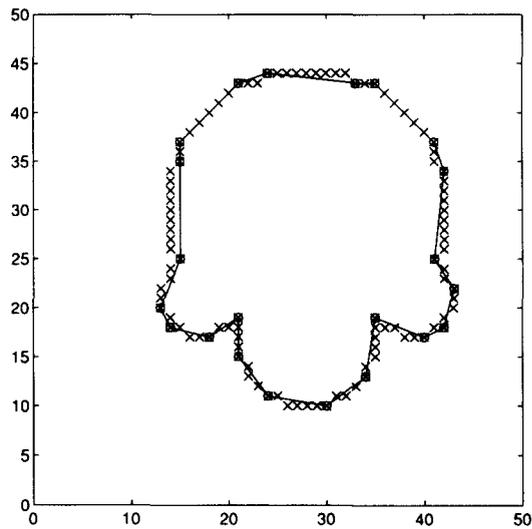


Fig. 4(b). Polygonal approximation of a curve with four semicircles using the Teh-Chin algorithm ($N = 102, M = 22$).

Table 1. Results of the present method and the Teh-Chin algorithm

Digital curve	Fig. 1	Fig. 2	Fig. 3	Fig. 4
Number of points	60	120	45	102
Results of the present method				
Number of dominant points	10	18	8	4
Integral square error	2.99	8.25	2.51	6.94
Results of the Teh-Chin algorithm				
Number of dominant points	15	29	13	22
Integral square error	7.20	14.96	5.93	20.61

curves based on this concept. Teh and Chin state their procedure as follows:

"The procedure first determines the region of support for each point based on its local properties, then computes measures of relative significance (e.g. curvature) of each point and finally detects dominant points by a process of nonmaxima suppression."⁽¹⁰⁾

Some experimental results of the Teh-Chin algorithm are shown in Figs 1(b), 2(b), 3(b) and 4(b). The vertices detected by the Teh-Chin algorithm are marked by small circles and the solid lines are the polygonal approximations. In these experiments, the k -cosine measure is selected as the measure of significance.

Some features of our algorithm and the Teh-Chin algorithm are tabulated in Table 1, including the number of dominant points and the integral square error. As shown in Table 1, our algorithm outperforms the Teh-Chin algorithm in the sense of approximation errors. Our algorithm results in smaller errors and requires fewer vertices than the Teh-Chin algorithm. The computational complexity of our algorithm is higher than that of the Teh-Chin algorithm. However, it is still in the acceptable range.

7. DISCUSSION

The number of vertices to be used can be given arbitrarily or selected by the algorithm automatically. For example, we can constrain the approximation error to be within some threshold per point. The dynamic programming algorithm is terminated at the value of M which first meets the constraint. In this way, the constructed curve will have a minimum number of vertices and acceptable level of approximation error.

If the digital curve is corrupted with noise, a preprocess will be required to smooth out the noise. The method proposed by Ansari and Huang⁽¹¹⁾ can be applied. In their algorithm, a smoothing procedure is proposed to reduce the noise effect based on the Gaussian filter.

8. CONCLUSIONS

In this paper the dynamic programming algorithm is applied to choose an optimal set of vertices. A simple approach is proposed to design circular arcs between each pair of adjacent vertices with low computational complexity. The curve-fitting method is generalized to approximate closed curves and fast algorithm is also provided. Our algorithm has been tested on a number of digital curves and satisfactory results have been obtained.

REFERENCES

1. B. K. Ray and K. S. Ray, Determination of optimal polygon from digital curves using L_1 norm, *Pattern Recognition* **26**(4), 505–509 (1993).
2. B. K. Ray and K. S. Ray, A non-parametric sequential method for polygonal approximation of digital curves, *Pattern Recognition Lett.* **15**, 161–167 (1994).
3. J. C. Perez and E. Vidal, Optimum polygonal approximation of digitized curves, *Pattern Recognition Lett.* **15**, 743–750 (1994).
4. G. Medioni and Y. Yasumoto, Corner detection and curve representation using cubic B-splines, *Comput. Vis. Graphics Image Process.* **39**, 267–278 (1987).
5. F. Lu and E. E. Milios, Optimal spline fitting to planar shape, *Sign. Process.* **37**, 129–140 (1994).
6. U. M. Landau, Estimation of a circular arc center and its radius, *Comput. Vis. Graphics Image Process.* **38**, 317–326 (1987).
7. S. M. Thomas and Y. T. Chan, A simple approach for the estimation of circular arc center and its radius, *Comput. Vis. Graphics Image Process.* **45**, 362–370 (1989).
8. M. Maes, On a cyclic string-to-string correction problem, *Inf. Process. Lett.* **35**, 73–78 (1990).
9. R. A. Wagner and M. J. Fischer, The string-to-string correction problem, *J. Ass. Comput. Mach.* **21**, 168–173 (1974).
10. C. H. Teh and R. T. Chin, On the detection of dominant points on digital curves, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(8), 859–872 (August 1989).
11. N. Ansari and K. Huang, Non-parametric dominant point detection, *Pattern Recognition* **24**(9), 849–862 (1991).

About the Author—SOO-CHANG PEI was born in Soo-Auo, Taiwan, China, on 20 February 1949. He received the B.S. degree from the National Taiwan University in 1970 and M.S. and Ph.D. degrees from the University of California, Santa Barbara, in 1972 and 1975, respectively, all in electrical engineering. He was an engineering officer in the Chinese Navy Shipyard at Peng Fu Island from 1970 to 1971 and a Research Assistant at the University of California, Santa Barbara, from 1971 to 1975. He was Professor and Chairman in the Department of Electrical Engineering at Tatung Institute of Technology from 1981 to 1983. Presently, he is the Professor and Chairman of the Department of Electrical Engineering at the National Taiwan University. His research interests include digital signal processing, digital picture processing, optical information processing, laser and holography. Dr Pei is a member of the IEEE, Eta Kappa Nu and the Optical Society of America.

About the Author—JI-HWEI HORNG was born in Kinmen, Republic of China, on 13 April 1968. He received the B.S. degree from the Tamkang University in 1990 and M.S. degree from the National Taiwan University in 1992. He is now at the National Taiwan University studying for a Ph.D. degree. His research interests include image analysis and pattern recognition.