# Parametric approximations of contours of digitized characters

R. Legault and C.Y. Suen

*Dept. of Computer Science*
*Concordia University*
*1455 de Maisonneuve West*
*Montreal, Quebec H3G 1M8*
*Canada*

**Abstract**

Contours of digitized characters are segmented by detection of sufficiently curved arcs. The resulting reference points are closely related to the structural features of characters (such as endpoints, cavities...). Piecewise approximation is done with parametric cubics and quartics and results are compared. Very good quality approximations are obtained at relatively low cost.

KEYWORDS: Contour segmentation, contour approximation, parametric curves, digitized characters.

## 1. Introduction

The goal of the present work is to find an efficient way to approximate contours in a piecewise manner. The analytic representation of contour pieces could then be used to derive information (such as curvature, areas etc...) related to the detection of features for character recognition. Such approximations could also be of use in many other applications.

We can identify the following desirable properties for our contour approximation scheme:

**Simplicity**. The partitioning of the contour should yield a *small number* of pieces, to be approximated by *low order* polynomials.

**Goodness of approximation.** Since digitization tends to yield wiggly contours, it is counter-productive to follow them exactly. However the approximation should be good enough to capture all features which play a useful role in recognition.

**Invariance**. Contour segmentation and quality of fit should be relatively independant of the resolution used for digitization.

It is also desirable that the contour partition itself bears a close relationship to the features to be subsequently detected.

Many approaches have been developed for contour approximation. The most common is probably polygonal approximation [2,5,9]. But parametric cubics have also been used [4,7]. Because the contours of handwritten characters normally present fairly long curved portions, we have opted for parametric cubics and quartics.

The initial problem to be solved is contour partitioning. We would like to obtain a limited number of reference points along the contour, which are related to the structural features of the character, and between which the selected curves will produce good approximations.

Several techniques exist for curve partitioning [6], detecting corners [1,3] or critical points [8]. But spurious corners are often obtained, while some true corners may go undetected. And critical points tend to occur in clusters. The more sophisticated corner detection algorithms reduce these defects but lose significantly in efficiency. Another difficulty is that corners — defined as points where we have a significant enough discontinuity in the mean curvature — are not the only points of interest here: we would also like to obtain "middle points" of large cavities for example. A new technique is developed to meet these needs.

After contour partitioning, each contour piece must be approximated. Optimal approximations are time consuming. Our approach instead is to find heuristics yielding generally good approximations.

## 2. Outline of the method and preliminary definitions

Each contour is extracted as a chain of consecutive points, circumscribing objects (body regions or holes) in the counter-clockwise direction. The main steps of the method are then

i) smoothing and initial processing of the contour;
ii) selection of reference points;
iii) parametric approximation of each piece of contour, between consecutive reference points.

We now introduce some definitions and notations:

a) Let the contour points be labelled from 1 through N.
b) Let [j,k] represent the sequence of consecutive contour points starting with point j and ending with point k. The same sequence, without j and k, is denoted by (j,k).
c) Let $C_j$ be the cumulated chord distance from point 1 to point j.
d) Let $\Theta_j$ be the direction change, in radians, at point j and defined as in figure 1 (positive in counter-clockwise direction).

$C_j$ and $\Theta_j$ are defined for $j\in[1,N]$ and $\Theta_j\in[-\pi,\pi]$.

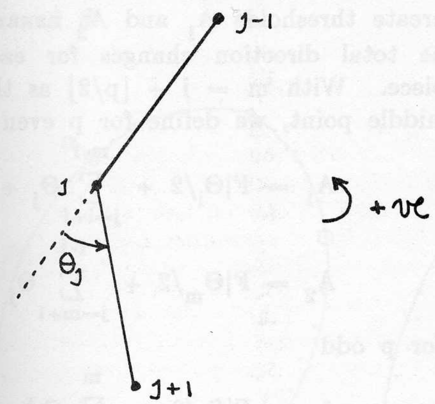


Figure 1: Direction change at point j

## 3. Smoothing and initial processing

The original list of contour points is first smoothed out. For every point j of the contour, we replace its coordinates by averaging them with the coordinates of the preceding and following contour points. This is performed twice for every point. The result is illustrated in figure 2.

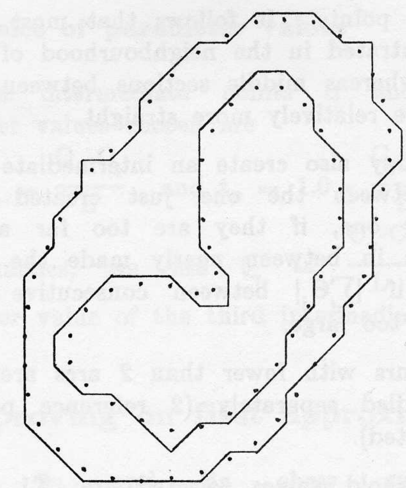

Figure 2: Contour points after smoothing

This operation removes insignificant wiggles and noise along the border of the digitized character. A smoother contour is obtained, which can be more meaningfully approximated by parametric polynomials (which themselves are smooth curves). However corners are rounded off, potentially reducing the distinction between similar characters such as 'Z' and '2' etc...

After smoothing, the values of $C_j$ and $\Theta_j$, as defined in the previous section, are obtained for each point.

## 4. Selection of reference points

**Definition.** An *arc* is defined as the longest sequence of contour points [k,k+l-1] satisfying the following 3 conditions:

i) $|\Theta_k| \geq T1$
ii) $\Theta_j$ has constant sign for $j\in[k,k+l-1]$
iii) $\Theta_{j-1}\Theta_j \geq T1$ for $j\in[k+1,k+l-1]$

For every arc, we obtain $\Theta sum = \sum\Theta_j$ and $\Theta max = \max |\Theta_j|$, $j\in[k,k+l-1]$. Let $j_{max}$ be the label of the point where $\Theta max$ is reached. Reference points are then selected as follows:

a) if $|\Theta sum| \geq T2$, a reference point is created within the arc. If $\Theta max > T3$, the reference point will be $j_{max}$ (this selects sharp corner points within an arc). Otherwise, we examine the middle point of the arc and its 2 neighbours on each side (5 points in all) and select the one for which $|\Theta_{j-1}+\Theta_j+\Theta_{j+1}|$ is maximum. This is the key rule which creates the vast majority of reference points. It follows that most curvature is concentrated in the neighbourhood of reference points, whereas middle sections between reference points are relatively more straight.

b) We may also create an intermediate reference point between the one just created and the preceding one, if they are too far apart and some arc in between nearly made the threshold T2, or if $|\sum\Theta_j|$ between consecutive reference points is too large.

c) Contours with fewer than 2 arcs are detected and handled separately (2 reference points are then created).

The threshold values selected are: T1 = 0.005; T2 = 0.5 (28.6°); and T3 = 0.75 (43.0°). The general procedure applied is to scan the Θ-array and look for the first negligible Θ-value. Then we start building up the first arc. The last arc will generally encompass points in the last portion of the contour and points in the initial portion as well. Thus the reference points obtained are independant of the initial point chosen for the contour. And except for some rare cases handled by rule b) above, they are also independant of the scanning resolution.

Figure 3 shows the location of the reference points for the inner and outer contours of 2 digits.

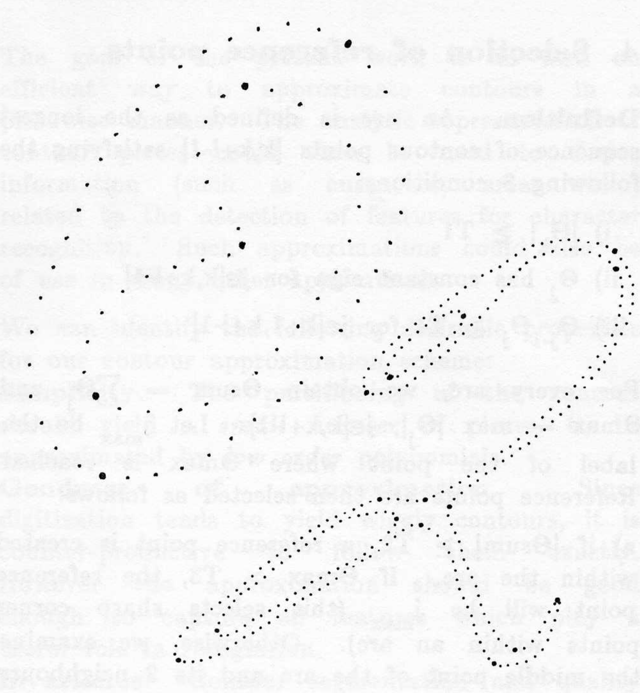

Figure 3: Reference points selected

## 5. Approximation for each contour piece

A contour piece is a sequence [i,f], where points i and f are consecutive reference points. Each contour piece will be approximated by a pair of parametric cubics/quartics:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \; (+ a_4 t^4)$$

$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \; (+ b_4 t^4),$$

where the terms in parentheses are only included for quartics. The parameter value is normalized such that, for each contour piece, t=0.0 for the initial point and t=1.0 for the final point.

For *cubics*, the linear system of equations will be fully determined if we also require the approximation to go through 2 intermediate points in (i,f), or else through 1 intermediate point with the additional condition of matching specified slopes at points i and f. The latter approach often gives very poor approximations, even for some short pieces, so we opt for the first approach.

### 5.1 Choice of intermediate points

For each contour piece, [i,f], we define its point count $p \equiv f-i$ and its chord length $L \equiv C_f - C_i$.

a) <u>Selection of intermediate points</u> (case $\Theta_i\Theta_f > 0$).

We create thresholds $A_1$ and $A_2$ as a fraction F of the total direction changes for each half of the piece. With $m = i + \lfloor p/2 \rfloor$ as the label of the middle point, we define for p even

$$A_1 = F\left|\Theta_i/2 + \sum_{j=i+1}^{m-1} \Theta_j + \Theta_m/2\right|$$

$$A_2 = F\left|\Theta_m/2 + \sum_{j=m+1}^{f-1} \Theta_j + \Theta_f/2\right|$$

and for p odd

$$A_1 = F\left|\Theta_i/2 + \sum_{j=i+1}^{m} \Theta_j\right|$$

$$A_2 = F\left|\sum_{j=m+1}^{f-1} \Theta_j + \Theta_f/2\right|.$$

The ***smallest*** k such that $\left|\Theta_i/2 + \sum_{j=i+1}^{k} \Theta_j\right| \geq A_1$ is the first intermediate point, labelled $\alpha$. The ***largest*** k such that $\left|\sum_{j=k}^{f-1} \Theta_j + \Theta_f/2\right| \geq A_2$ yields the second intermediate point, labelled $\beta$.

However we always require that ($\alpha$-i), (f-$\beta$) $\in$ (Min,Max). Finally for *quartics*, the third intermediate point, labelled $\gamma$, would be halfway between $\alpha$ and $\beta$.

For cubics, we use F=0.8, Min=p/5 and Max=p/3. Thus 80% of the curvature in a given contour piece is concentrated between the endpoints and the intermediate points. And at most 20% is left for the middle section, between the intermediate points. For quartics, F=0.6, Min=p/6 and Max=p/4.

b) Selection of intermediate points (case $\Theta_i\Theta_f<0$).

Here we maintain constraints about the location of $\alpha$ and $\beta$. For cubics, Min=p/6 and Max=p/3 whereas for quartics we use Min=p/7 and Max=p/5.

The rules for obtaining $\alpha$ are as follows:

i) set $\alpha$=i;

ii) increment $\alpha$ by 1, as long as $\Theta$ is in the same direction and has not reached i+Max;

iii) skip over points with negligible $\Theta$ and branch back to ii if the first non-negligible $\Theta$ is of the same sign as $\Theta_i$;

iv) skip over pairs of consecutive points with $\Theta$-values equal but opposite in sign.

To find $\beta$, we proceed likewise, starting from point f and moving backwards. For quartics, the third intermediate point is again halfway between $\alpha$ and $\beta$.

### 5.2 Choice of parameter values

For the intermediate points $\alpha$ and $\beta$, the parameter values chosen are

$$t_\alpha = \frac{C_\alpha - C_i}{L}, \text{ and } t_\beta = 1.0 - \frac{C_f - C_\beta}{L}.$$

For quartics, we use $t_\gamma = \frac{C_\gamma - C_i}{L}$ as the parameter value of the third intermediate point.

## 6. Improving on first approximation

Figures 5a and 6a show some first approximations obtained with *cubics*. Typically, the fit is not very good; and the slopes of the 2 pieces connecting at a reference point may not match well, resulting in pointed endings. For quartics, similar results are observed, but the mid-section fit is better, due to the third intermediate point.

However improved approximations result from modifying the parameter values at the intermediate points as follows:

$$t_\alpha = f_\alpha \frac{C_\alpha - C_i}{L}, \text{ and}$$

$$t_\beta = 1.0 - f_\beta \frac{C_f - C_\beta}{L}$$

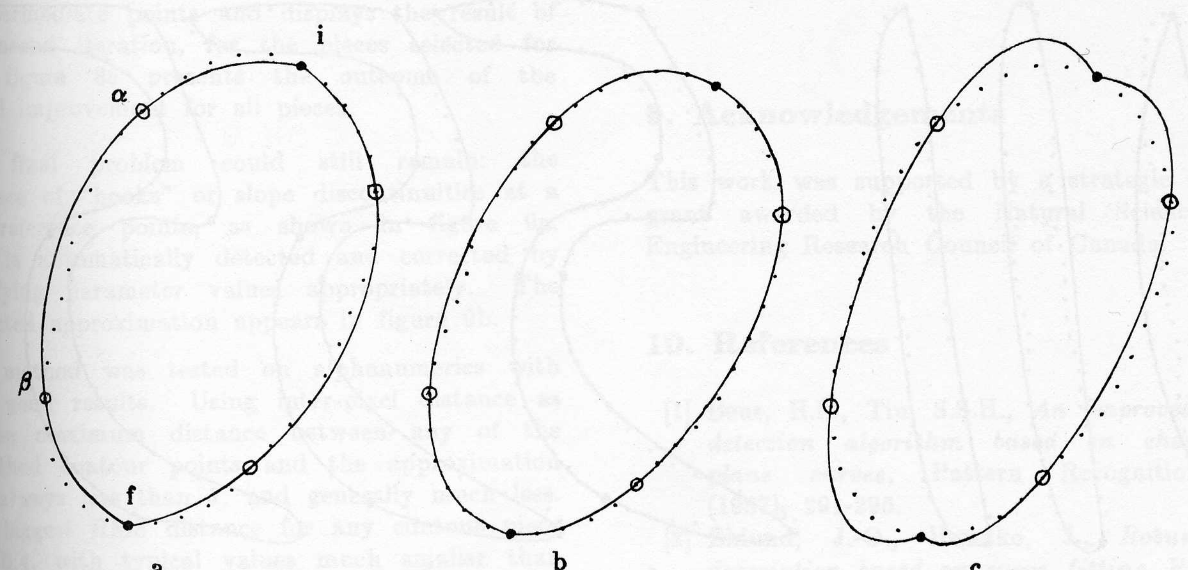


Figure 4: Modifying parameter values at $\alpha$ and $\beta$

a) $f_\alpha = f_\beta = 1.00$

b) $f_\alpha = f_\beta = 1.10$

c) $f_\alpha = f_\beta = 1.25$

The effect of the multipliers $f_\alpha$ and $f_\beta$ is illustrated in figure 4. Figure 4a shows 2 contour pieces (for which $\Theta_i\Theta_f>0$) with $f_\alpha=f_\beta=1.0$. Usually, the approximation is ***inside*** the contour points in $[i,\alpha]$ and $[\beta,f]$ and ***outside*** the contour points in $[\alpha,\beta]$. Figures 4b and 4c show what is obtained when we set $f_\alpha=f_\beta=1.1$ and $f_\alpha=f_\beta=1.25$ respectively. As we increase $f_\alpha$ and $f_\beta$, the curvature in $[i,\alpha]$ and $[\beta,f]$ is increased while the $[\alpha,\beta]$ portion of the curve becomes flatter.

A unique value for the multipliers does not give good results for all pieces. We now describe how an appropriate $f_\alpha$ is derived for a specific piece. $f_\beta$ is found in a similar way.

First, we obtain the approximation with parameter values as described in the previous section. Using a maximum of 3 testpoints in $(i,\alpha)$, we obtain their average distance D to the line segment from i to $\alpha$. Now the ratio $D/(C_\alpha-C_i)$, raised to some appropriate fractional power, provides us with CV, a "measure of curvature" for the $[i,\alpha]$ portion of the contour piece.

Secondly, for each testpoint, we compute the ratio of its distance to the approximating cubic (or quartic) to its distance to the line segment from i to $\alpha$. Averaging these ratios, we obtain the "fraction of curvature still missing", which we denote by FM. Using CM = CV.FM, a first estimate for $f_\alpha$ is

$$f_\alpha = 1.0 + k.CM$$

where k=1.0 for contour pieces with $\Theta_i\Theta_f>0$, and has value 1.2 (cubics) or 2.0 (quartics) for other pieces.

Additional adjustments are needed in particular situations. For contour pieces with $\Theta_i\Theta_f>0$, CM is increased when curvature is particularly concentrated near the reference point and has a large value. For cubics, the smallest of $f_\alpha$ and $f_\beta$ often needs to be reduced.

Figures 5b and 6b show some good results obtained by improving the first approximations for ***cubics***. Figures 5c and 6c present the improved first approximations for ***quartics***. As expected, quartics provide a better fit, especially if there is more than 1 inflection point in $[i,f]$.

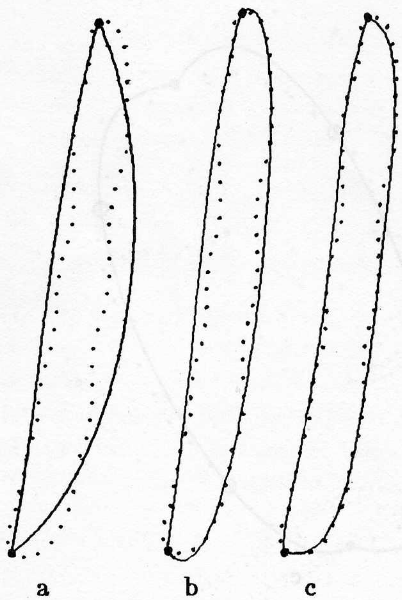


Figure 5: Approximation results for a '1'

a) first fit with cubics
b) improved fit with cubics
c) improved fit with quartics

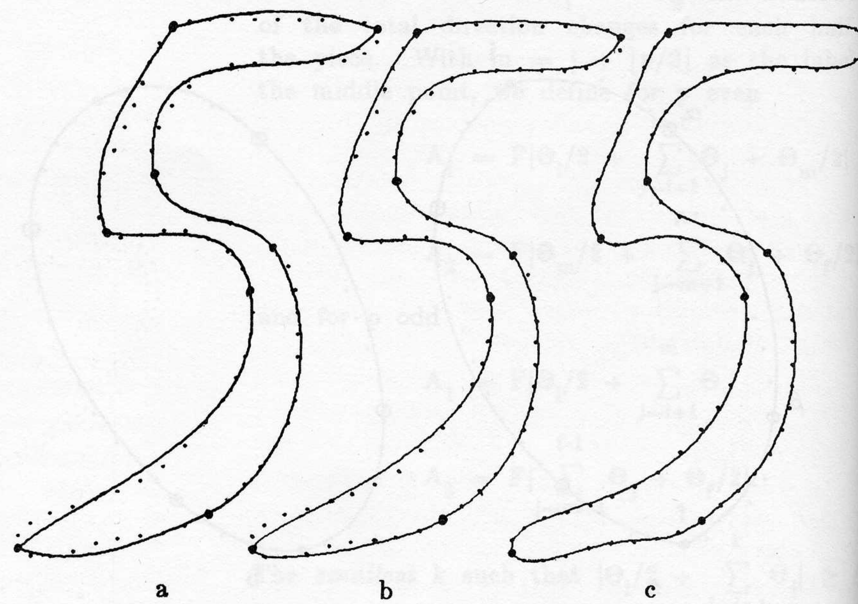


Figure 6: Approximation results for a '5'

a) first fit with cubics
b) improved fit with cubics
c) improved fit with quartics

## 7. Fit estimation and more improvements

Some results are not always as good as in figures 5 and 6. We consider how to improve our method further. Since quartics offer more flexibility and the cost of obtaining them with our scheme is very similar to that of cubics, we restrict our attention to quartics.

A flowchart of the complete scheme to approximate one contour piece with quartics is shown in figure 7. After obtaining the first improved approximation, we measure the fit in each of the sub-intervals $(i,\alpha)$, $(\alpha,\gamma)$, $(\gamma,\beta)$ and $(\beta,f)$. If it is not satisfactory, some or all of the intermediate points are displaced slightly. A new iteration is then performed (sections 5.2 and 6) and its fit in each of the 4 sub-intervals is also evaluated. Results are compared and the best iteration is kept.

The best iteration (or simply the first, if no second iteration was deemed necessary) goes through a second improvement stage, very similar to what was described in section 6, where the parameter values $t_\alpha$, $t_\gamma$ and $t_\beta$ are again possibly modified. However corrections this time are limited to a maximum of 50% of the initial corrections.

Figure 8 provides an illustration of the process: figure 8a shows the first improved approximation for an 'S'; figure 8b indicates the displacement of intermediate points and displays the result of the second iteration, for the pieces selected for this; figure 8c presents the outcome of the second improvement for all pieces.

One final problem could still remain: the presence of "hooks" or slope discontinuities at a few reference points, as shown in figure 9a. This is automatically detected and corrected by modifying parameter values appropriately. The corrected approximation appears in figure 9b.

The method was tested on alphanumerics with very good results. Using inter-pixel distance as 1, the maximum distance between any of the smoothed contour points and the approximation was always less than 1, and generally much less. The biggest RMS distance for any contour piece was 0.4, with typical values much smaller than this. On the average, approximately 30% of contour pieces required a second iteration, which turned out to provide a better fit than the first iteration 4 times out of 5.

A program, written in FORTRAN, accepts already smoothed contours as its input. For 16 characters (1296 contour points; 26 contours; 125 pieces to approximate), it required 0.34s to obtain the reference points and 2.47s to derive the approximations, when the program was run on a CDC Cyber 830D computer.

## 8. Concluding remarks

Piecewise approximation of contours of digitized characters with parametric cubics and quartics was investigated. The contours were first partitioned at reference points, whose position is closely related to structural features such as endpoints, cavities etc... Heuristics were found to ensure a very good approximation at relatively low cost.

It appears the choice of reference points could be improved by relating them more closely to the detection of features. We could then possibly be satisfied with coarser approximations obtained more efficiently, as long as they retain all relevant features for recognition. This would also result in fewer reference points in many cases.

We are currently working on character recognition methods that would use the reference points as "feature points". In this context, the approximation scheme presented here could be used selectively for certain contour pieces for which more detailed information is required.

## 9. Acknowledgements

This work was supported by a strategic research grant awarded by the Natural Sciences and Engineering Research Council of Canada.

## 10. References

[1] Beus, H.L., Tiu, S.S.H., *An improved corner detection algorithm based on chain-coded plane curves*, Pattern Recognition, **20**, (1987), 291-296.

[2] Eklund, J.-O., Howako, J., *Robust shape description based on curve fitting*, Proc. 7th Int. Conf. on Pattern Recognition, Montreal, (1984), 109-112.

[3] Freeman, H., Davis, L.S., *A corner finding algorithm for chain-coded curves*, IEEE Transactions on Computers, **C-26**, (1977), 297-303.

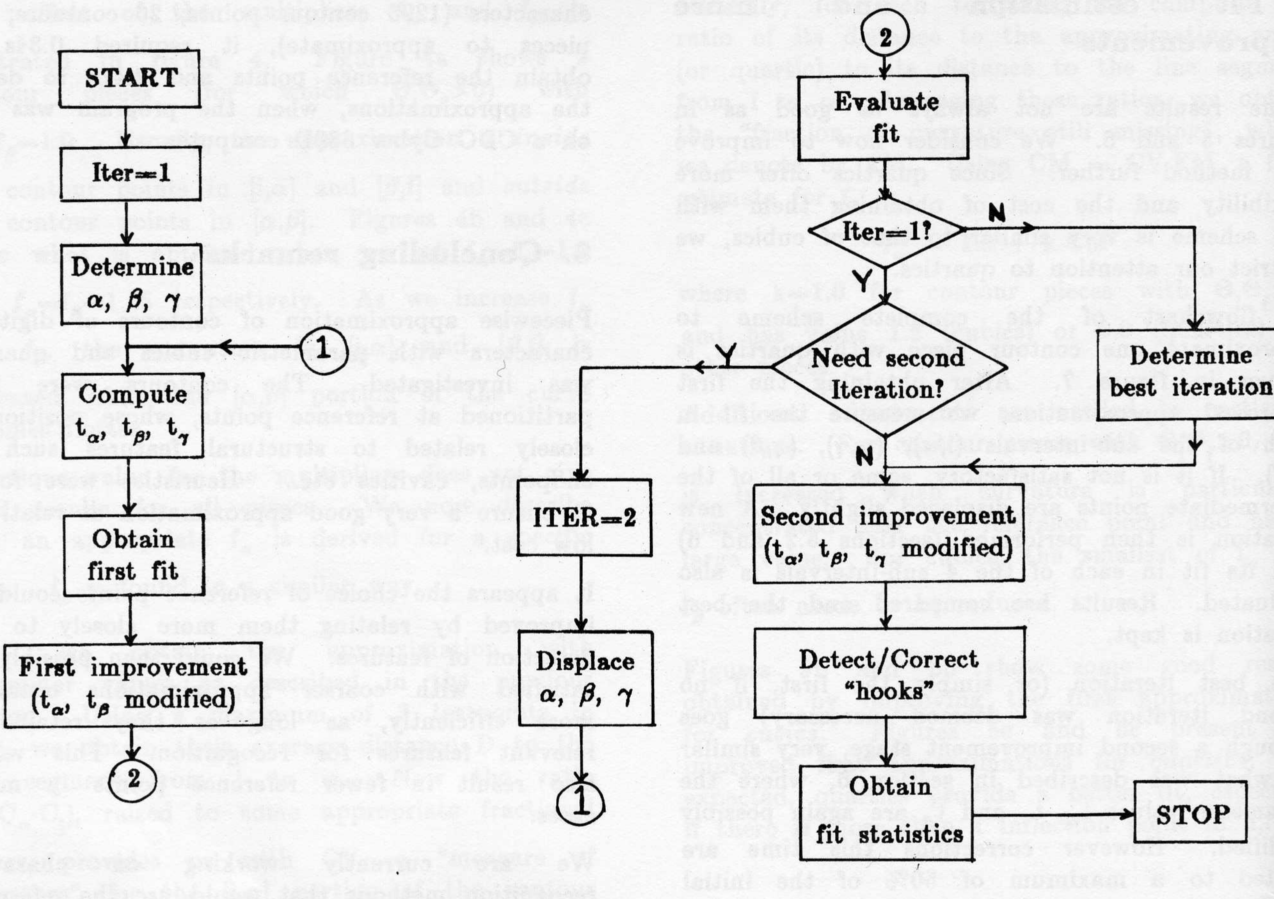


**Figure 7: Complete flowchart of approximation with quartics**

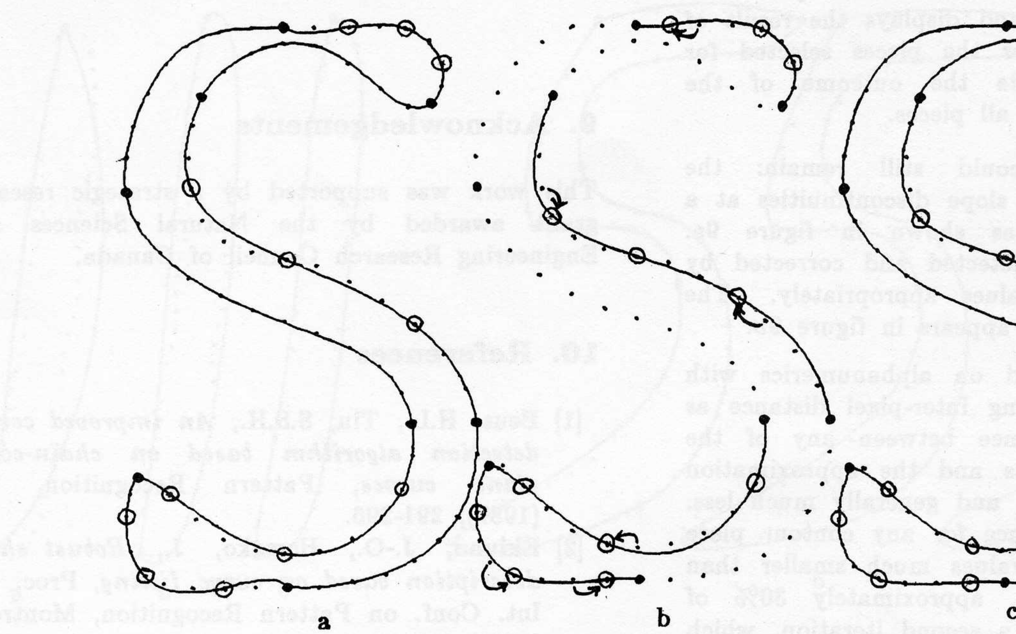


**Figure 8: Different stages of approximation with quartics**

**a) result of first improvement**
**b) second iteration for selected pieces**
**c) result of second improvement**

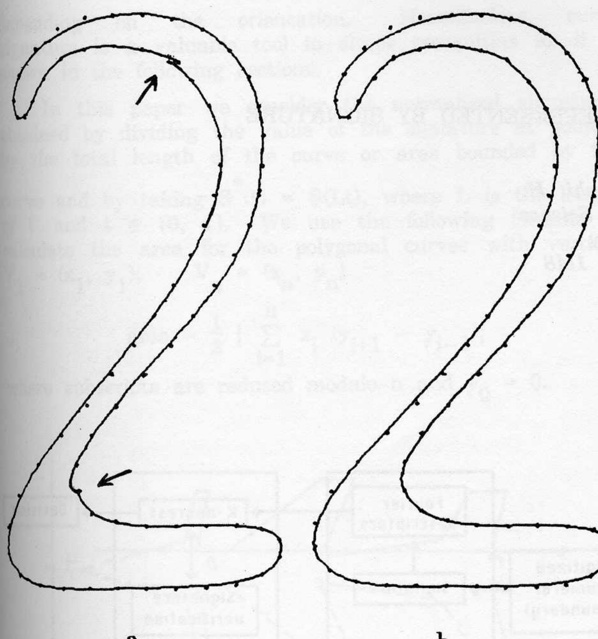


Figure 9: Correction for "hooks"


[4] Langridge, D.J., *Curve encoding and the detection of discontinuities*, Computer Graphics and Image Processing, **20**, (1982), 58-71.

[5] Pavlidis, T., *Curve fitting as a pattern recognition problem*, Proc. 6th Int. Conf. on Pattern Recognition, Munich, (1982), 853-859.

[6] Phillips, T.-Y., Rosenfeld, A., *A method of curve partitioning using arc-chord distance*, Pattern Recognition Letters, **5**, (1987), 285-288.

[7] Rom, M., Bergman, S., *A new technique for automatic contouring and contour representation from machine-readable data*, The Computer Journal, **29**, (1986), 467-471.

[8] Tejwani, Y.J., Jones, R.A., *Determination of critical points on planar shapes using an adaptive line of sight method*, Proc. 7th Int. Conf. on Pattern Recognition, Montreal, (1984), 113-115.

[9] Wall, K., Danielsson, P.-E., *A fast sequential method for polygonal approximation of digitized curves*, Computer Vision, Graphics and Image Processing, **28**, (1984), 220-227.