An Application Based Automatic Outline Capturing of Images

Presented to Dr Muhammad Sarfraz
By
Syed Adnan Yusuf
Syed Zeeshan Muzaffar

Advanced Graphics (ICS 535)
(Semester 022)

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

Automatic Outline Capturing For Open Curves

Abstract. *The paper presents a system that automatically captures the outline of digitized images. The elementary purpose of this system is to detect the significant points of the open curves that result in an optimal fit. The system can also be used to capture the outline of the images that represent closed contours. We tested our system on various application domains such as font regeneration, hand drawn images, scientific and mathematical functions. A method to handle curves with open contours is given.*

Furthermore, the results are tested for two out of six edge detectors implemented in our system and are compared for their optimality in curve fitting. The objective is to find the minimum number of significant points by keeping in mind the optimization of closeness of the fit between the original digitized curve and the parametric curve. To achieve this 5 different cases are presented by incorporating Lease Square Method with the General Hermite Cubic Function.

Keywords: Open Curves, Image Outline, Curve Fitting.

1. Introduction

Continuous data actually does not exist, for when we digitize an outline or curve; we actually sample discrete points along a curve. We can then fit a mathematical function to determine some curve, and characterize it by parameters which can come as close to the sampled points as we please.

Curve designing has been one of the significant problems of Computer Graphics. There are number of applications where finding a mathematical curve description of the desired shape is beneficial. Curves that provide closeness and robustness in the actual fit are of much interest. The ideas such as end point interpolation, detection of characteristic points, least square method, recursive subdivision and parameterization can be used for curve fitting.

2. Theory and Application

Curve Regeneration

The technique of curve fitting finds its application in a lot of application domains such as font regeneration, hand drawn images, scientific and mathematical functions. The overall process of curve regeneration goes as follows:

1. Image Capturing

Features of an image are captured using some input device such as a digital camera, scanner, light pen, etc. The image involves pre-processing making it appropriate for boundary extraction. The step involves Conversion of an RGB image to Binary, Noise Removal, Digitization and Chain code extraction [1] [2]. Chain codes are a notation for recording the list of edge points along a contour. The chain code specifies the direction of a contour at each edge in the edge list as shown in figure 1.

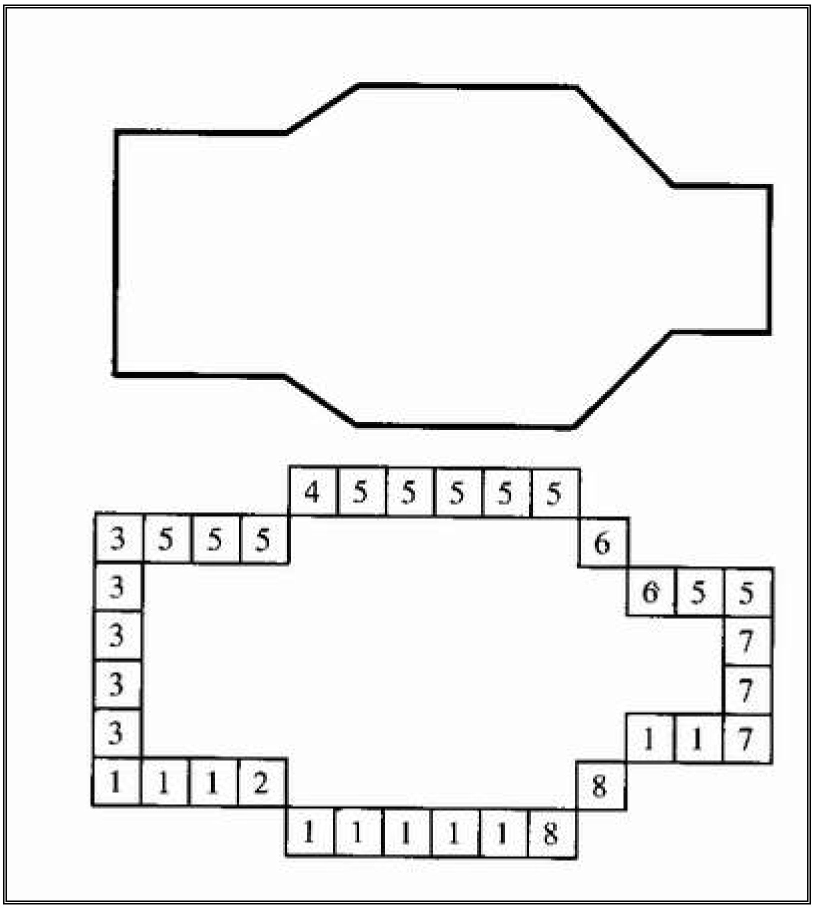


Figure 1: A curve and its chain code

2. Boundary Extraction

The Step involves extraction of significant curvature information from the noise free, digitized and chain coded curve. This results in a set of ordered points whose selection is being made on the basis of their corner strength [3] and contour fluctuations [4]. This might result in a group of points representing a single contour redundantly as shown in figure 2.

3. Curve Fitting

Conventionally, a number of curve fitting techniques are used [5]. Any fitting algorithm must address the issue of

- Finding an optimal method to fit the curve to the edges
- Method for measuring the closeness of the curve fit

Basically, four models are used for the curve fitting. Line segments are used to fit the edges between the set of points that represent a contour. This method can be used where the contours can be easily defined by joining the line segments. Circular Arcs can easily handle the problem which the Line segment model fails to overcome due to such shapes that are not the composition of line segments only. The model has its limitations when a contour is composed fully/partially of line segments. Conic Sections further elaborate the problem by providing the flexibility of using hyperbolas, parabolas and ellipse in the curve fit. The model has the same inherent disadvantage as with circular arcs.

A great degree of control can be achieved by using Cubic polynomial in the curve fit. Cubic polynomials are most often used because lower-degree polynomials give too little flexibility in controlling the shape of the curve.

Selection of these 4 models produces a trade-off between the complexities of implementation to the sharpness of fit. Higher degree polynomials can introduce unwanted wiggles and also require more computation. No lower-degree representation allows a curve segment to interpolate two specified endpoints with specified derivatives at each point.

4. Detection of Significant Points.

The Set of points obtained in Step 2 are candidates to a higher order of point category known as the significant points. The selection is being on the bases of the value of threshold set by a potential user. The points, thus obtained, represent a curve using minimal information and can be used in later stages to generate a similar curve using various curve fitting techniques.[5]

5. Checking for curve optimality.

There are several measures of goodness of fit of a curve to the candidate edge points [4] [5]. All of them depend on the error between the fitted curve and the candidate points forming the curve. Some commonly used methods are as follows:

- Maximum Absolute Error

It measures how much a point deviate from the curve in the worst case:

$$MAE = \max_i |d_i|$$

- Mean Squared Error

It gives an overall measure of the deviation of the curve from the edge points:

$$MSE = \frac{1}{n}\sum_{i=1}^{n} d_i^2$$

- Normalized Maximum Error

It is the ratio of the maximum absolute error to the length of the curve:

$$\varepsilon = \frac{\max_i |d_i|}{S}$$

- Ratio of the curve length to end point distance

It is a good measure of the complexity of the curve.

The methodology adopted assumes that sometimes corners are not detected precisely and only corner points are not sufficient to fit the curve which represents the original image. Therefore, some more points are needed to achieve a best fit. These points are called the *break points* and are used along with corner points to achieve the best fit by using the *least square method*. The subdivision methodology is used to conquest the desired solution. Another major difference lies in the curve model for the description of design curve. The outline capturing technique, instead of traditional Hermite cubic, is based upon a cubic model which has attracting features to control the curve segments.

6. Repeating Step 4 and 5 until an optimal result is obtained.

3. Corner Detection and Curve Fitting Algorithms

Various corner detection algorithm are proposed and tested. The curve regeneration process uses the IPAN algorithm [3] for the detection of high curvature points. The algorithm [3] works by assigning the measure of corner strength ('cornerity') to the high curvature points. It is a two pass algorithm. A curve is represented be a sequence of points P_i in the image plane. The ordered points are densely sampled along the curve but no regular spacing between them is assured.

First Pass:

In the first pass, in each curve point P the detector tries to inscribe in the curve a variable triangle (P^-, P, P^+) constrained by a set of simple rules.

$$d_{\min}^2 \le \left|P - P^+\right|^2 \le d_{\max}^2$$

$$d_{\min}^2 \le \left|P - P^-\right|^2 \le d_{\max}^2$$

$$\alpha \le \alpha_{\max},$$

Where $\left|P - P^+\right| = |a| = a$ is the distance between P and P^+, $\left|P - P^-\right| = |b| = b$ the distance between P and P^-, and $\alpha \in [-\pi, \pi]$ the opening angle of the triangle as shown in figure 2. The later is computed as

$$\alpha = \arccos \frac{a^2 + b^2 - c^2}{2ab}$$

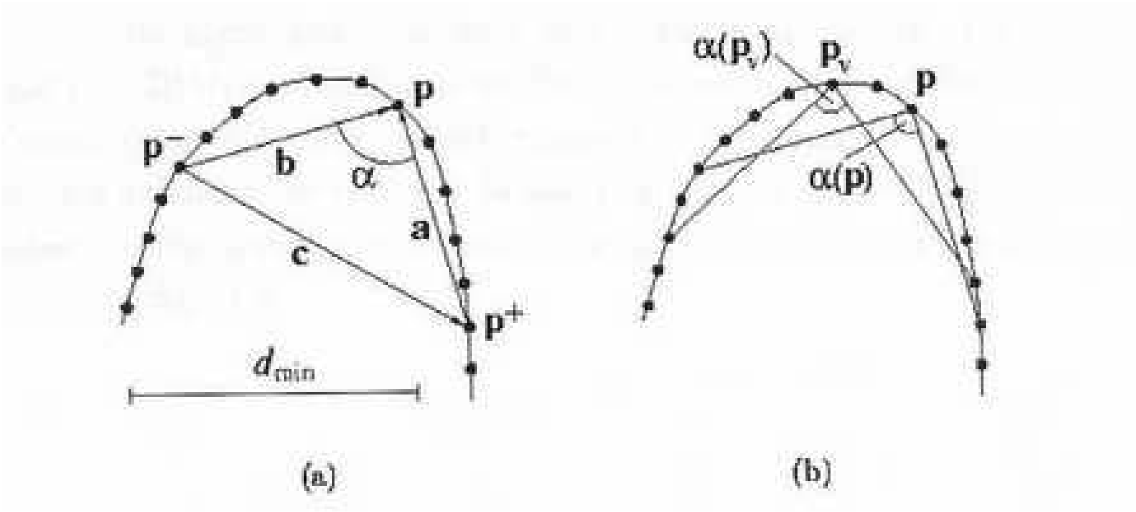
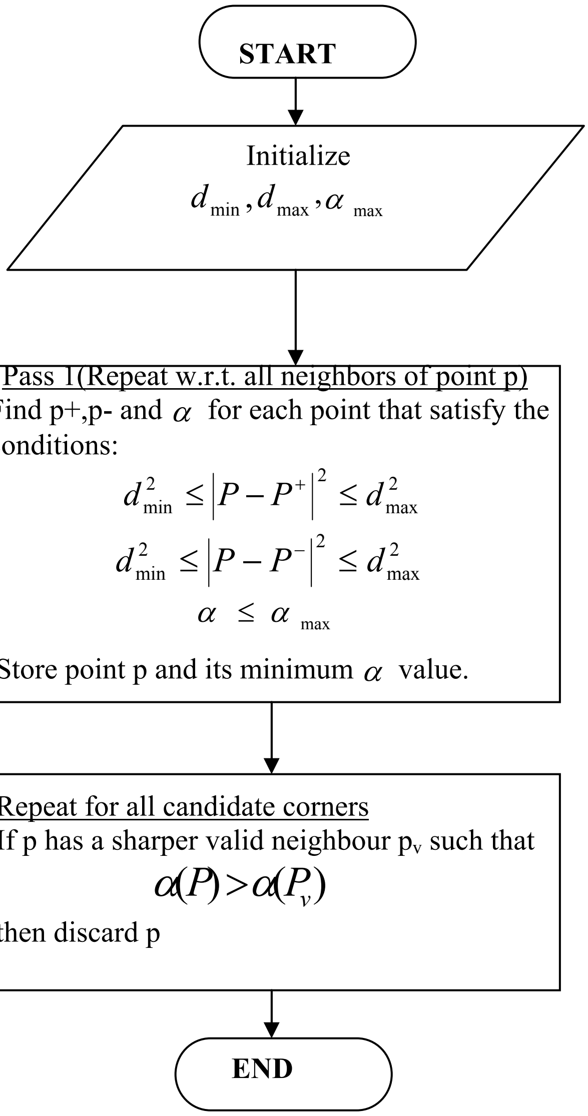


Figure 2: Detecting high curvature points. (a) Determining if P is a candidate point. (b) Test P for sharpness.

Second Pass

A corner detector can respond to the same corner in a few consecutive points. A step is needed to discard the non-maxima points. A candidate point P is discarded if it has a sharper valid neighbour P_v such that $\alpha(P) > \alpha(P_v)$. A candidate point is a valid neighbour of P if $|P - P_v|^2 \leq d_{\min}^2$ or the points adjacent to P.



4. Curve Fitting Optimization

To achieve an optimal design curve, we have following 5 cases. [6]

Case 1: $v_i = w_i = 1$ and tangents are measured by distance based choice for tangent vectors.

Case 2: $v_i = w_i$ and tangents are measured by distance based choice for tangent vectors.

Case 3: $v_i \neq w_i$ and tangents are measured by distance based choice for tangent vectors.

Case 4: $v_i = w_i = 1$ and tangents are calculated using Least Square Estimation.

Case 5: v_i, w_i and D_i are calculated using Least Square Estimation.

5. Results and Discussions

Our main objective in the implementation was to fit the curve on the set of data points that represent open contours. The system works for closed contours as well. These points can be obtained by a direct user input or through some pattern recognition technique. Option is given to test a user defined input or the output of a mathematical function if it complies with the standard input format given to the system.

The system was tested using 6 edge detection techniques using Arabic, English and User defined images. The edge detection techniques were as follows:

1. Sobel
2. Prewitt
3. Roberts
4. Zero-cross
5. Log
6. Canny

The boundaries extracted using these edge detectors resulted in open and closed curves. The system was developed to encounter both open and closed curves.

Results on Open and Closed Curves

The Curve Fitting Algorithm when applied over open and closed contours, displayed results in passes. The test results for Arabic Character 'Jeem' were as follows:

1.1.1 Closed Curve

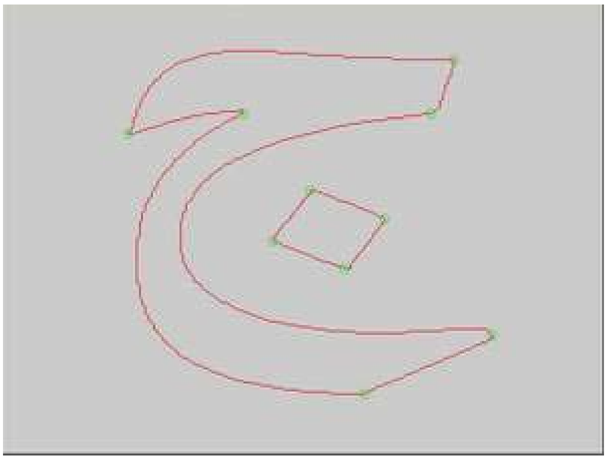


Figure 3: Outline of Arabic Character 'Jeem'. The character was tested with dmin = 7, dmax = 9, alphamax =150 using Sobel edge detection under case 4.

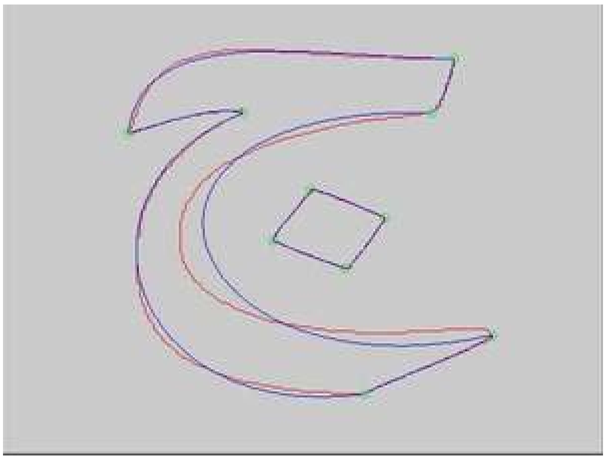


Figure 4: First iteration (Threshold = 2)

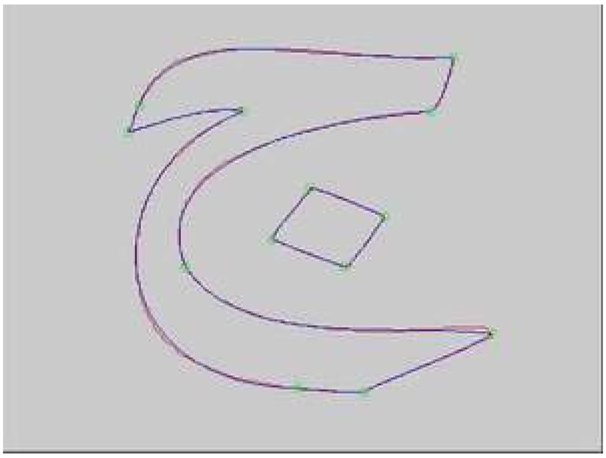


Figure 5: Second Iteration (Threshold = 2)

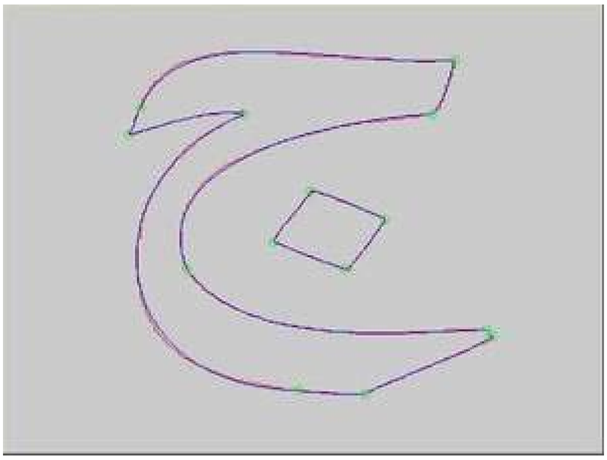


Figure 6: Final Iteration

1.1.2 Open Curve (User- Defined data)

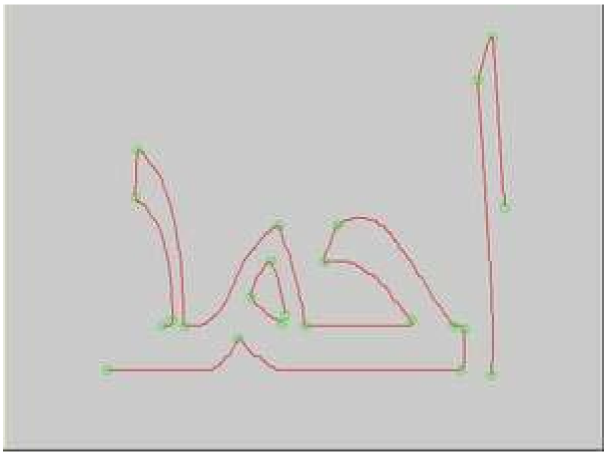


Figure 7: Outline of Arabic word 'Ahmad'. The character was tested with dmin = 7, dmax = 9, alphamax =150 using Sobel edge detection under case 4.

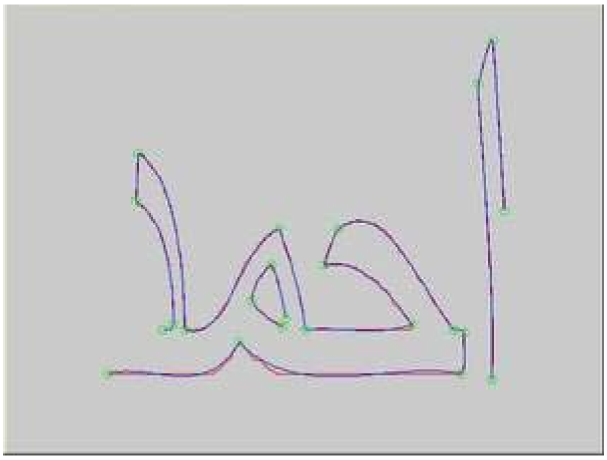


Figure 8: First iteration (Threshold = 2)

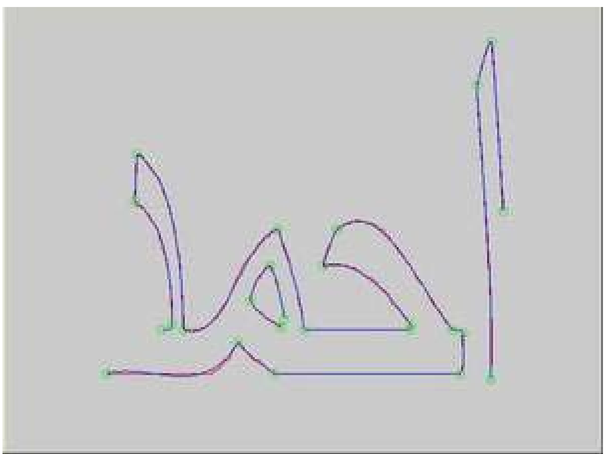
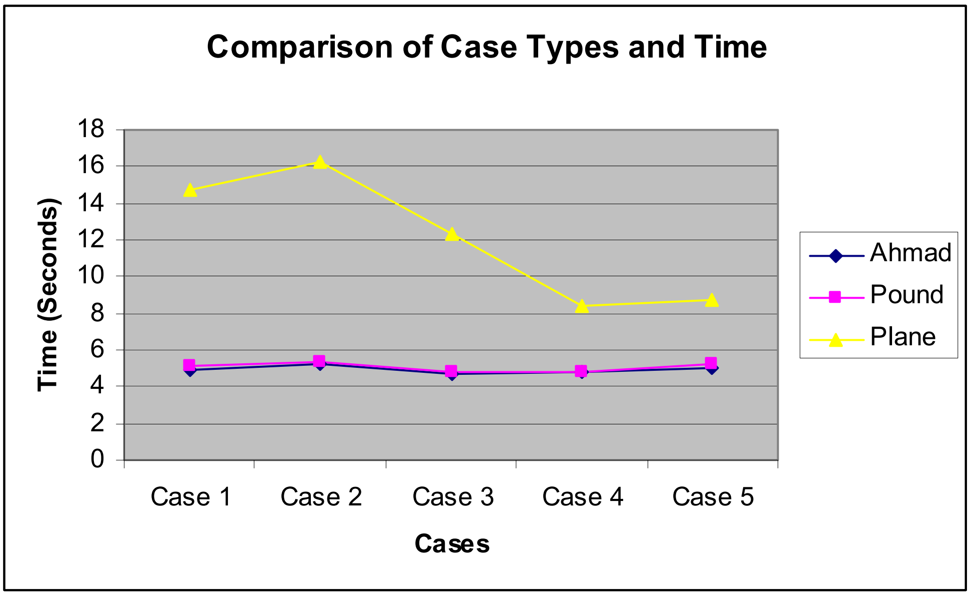
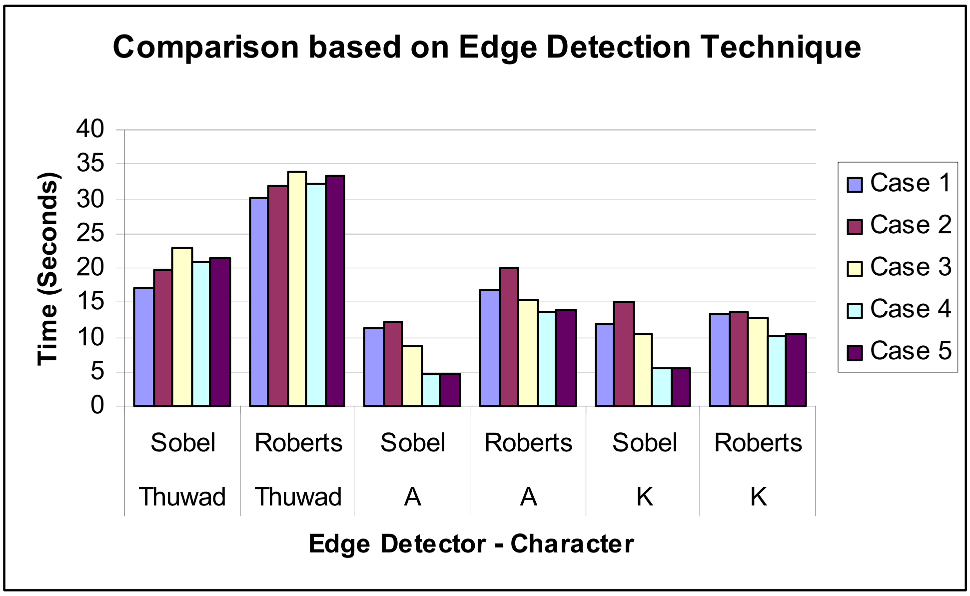


Figure 9: Final Iteration (Threshold = 2)

1.1.3 Comparisons based on Cases and Edge Detection Techniques used



All the 5 cases for Generalized Hermite Interpolant were tested on 3 different files using Sobel based edge detection. According to the results shown Case 4 ($v_i = w_i = 1$, Tangents calculation by Least Square Methods) the Results showed a more effective outcome when the size of file was large.



The comparison of various open and closed characters showed a Gaussian spread of values. Showing best results under Case 4 ($v_i = w_i = 1$, Tangents calculation by Least Square Methods). Furthermore, testing Roberts Edge Detector on a number of characters resulted in open contours.

4. Conclusions and Future Work

The system was tested over a number of open and closed contours with different edge detection techniques. The results showed an affective outcome using Sobel edge detection technique with Case 4. An interface was designed to log testing results more effectively.

We plan to detect significant points by estimating contour fluctuations [4]. We also plan to implement a neighborhood growing procedure to eliminate redundant significant point over a curvature. Furthermore, we plan to handle multiple joining edges over a curve [7].

.

References

[1] H. Freeman, L. S. Davis, *A corner finding algorithm for chain coded curves*, IEEE Trans Comput, C-24 (1975) 940-941.

[2] H-C. Liu, M. D Srinath, *Corner Detection from Chain chain code,* Pattern Recognition, 23, (1990) 51-68.

[3] D. Chetrikov., S. Zsabo, *A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Vurves*, Proc. 23rd Workshop of the Austrailian Pattern Recognition Group, 1999, 1751-184.

[4] N. Richard, T. Gilbert, *Extraction of Dominant Points by estimation of the contour fluctuations,* Pattern Recognition (35), 2002, 1447-1462.

[5] Jain, R., Kasturi, R, Schunk, B. G., *Machine Vision*, Page 192-209.

[6] Sarfraz, M., Faisal M. *,A Web Based Automatic Outline Capturing of Images,* Master Thesis.

[7] Hou Z. J., Wei G. W., *A new approach to edge detection,* Pattern Recognition 35 (2002) 1559-1570.