Capturing Image Outlines using Spline Computing Approach

Muhammad Sarfraz and Muhammad IhsanulHaq Sarfraz
Department of Information Science, Kuwait University
Adailiya Campus, P.O. Box 5969, Safat 13060, Kuwait
e-mail: prof.m.sarfraz@gmail.com

***Abstract*— An optimization technique is proposed for the outline capture of planar images. The overall technique has various phases including extracting outlines of images, detecting corner points from the detected outline, and curve fitting. Linear and conic interpolation methods have been used for vectorization. The idea of multilevel coordinate search has been used to optimize the shape parameters in the description of the generalized conic spline. The two methods ultimately produce optimal results for the approximate vectorization of the digital contours obtained from the generic shapes.**

Keywords- Optimization, corner points, generic shapes, curve fitting, spline

I. INTRODUCTION

Capturing and vectorizing outlines of images is one of the important problems of computer graphics, vision, and imaging. Various mathematical and computational phases are involved in the whole process. This is usually done by computing a curve close to the data point set. Computationally economical and optimally good solution is an ultimate objective to achieve the vectorized outlines of images for planar objects.

Curve modeling [21-23] plays significant role in various applications. The representation of planar objects, in terms of curves, has many advantages. For example, scaling, shearing, translation, rotation and clipping operations can be performed without any difficulty. Although a good amount of work has been done in the area [8-20], it is still desired to proceed further to explore more advanced and interactive strategies. Most of the up-to-date research has tackled this kind of problem by curve subdivision or curve segmentation.

This work is a presentation of two approaches using linear and conic interpolations. The linear interpolant approach is straight forward. However, the conic approach is inspired by an optimization algorithm based on multilevel coordinate search (MCS) by Huyer and Neumaier [24-25]. It motivates the authors to an optimization technique proposed for the outline capture of planar images. In this paper, the data point set represents any generic shape whose outline is required to be captured. We present an iterative process to achieve our objective. The algorithm comprises of various phases to achieve the target. First of all, it finds the contour of the gray scaled bitmap image [26-27]. Secondly, it uses the idea of corner points [1-7] to detect corners. These phases are considered as preprocessing steps. The next phase detects the corner points on the digital contour of the generic shape under consideration. Linear and conic interpolants are then used to vectorize the outline. In case of poor approximation, the insertion of intermediate points is made as long as the desired approximation or fit is achieved. The idea of multilevel coordinate search (MCS) is used to fit a conic spline which passes through the corner points. It globally optimizes the shape parameters in the description of the conic spline to provide a good approximation to the digital curve.

The organization of the paper is as follows, Section 2 discusses about preprocessing step which includes finding the boundary of planar object and detection of corner points. Section 3 is about the interpolant form of linear and conic spline curves. Overall methodology of curve fitting is explained in Section 4, it includes the idea of knot insertion as well as the algorithm design for the proposed vectorization scheme. Demonstration of the proposed scheme is presented in Section 5. Finally, the paper is concluded in Section 6.

II. PREPROCESSING

The proposed scheme starts with first finding the boundary of the generic shape and then using the output to find the corner points. The image of the generic shape can be acquired either by scanning or by some other mean. The aim of boundary detection is to produce an object's shape in graphical or non-scalar representation. Chain codes [27], in this paper, have been used for this purpose. Demonstration of the method can be seen in Figure 1(b) which is the contour of the bitmap image shown in Figure 1(a).

Corners in digital images give important clues for the shape representation and analysis. These are the points that partition the boundary into various segments. The strategy of getting these points is based on the method proposed in [1]. The demonstration of the algorithm is made on Figure 1(b). The corner points of the image are shown in Figure 1(c).

III. CURVE FITTING AND SPLINE

The motive of finding the corner points, in Section 2, was to divide the contours into pieces. Each piece contains the data points in between two subsequent corners inclusive. This means that if there are *m* corner

978-0-7695-3959-1/09 $26.00 © 2009 IEEE
DOI 10.1109/SITIS.2009.30

points cp_1, cp_2, ..., cp_m then there will be m pieces pi_1, pi_2, ..., pi_m. We treat each piece separately and fit the spline to it. In general, the i^{th} piece contains all the data points between cp_i and cp_{i+1} inclusive. After breaking the contour of the image into different pieces, we fit the spline curve to each piece. To construct the parametric spline interpolant on the interval $[t_0, t_n]$, we have $F_i \in R^m$, $i = 0,1,......,n$, as interpolation data, at knots t_i, $i = 0,1,......,n$.

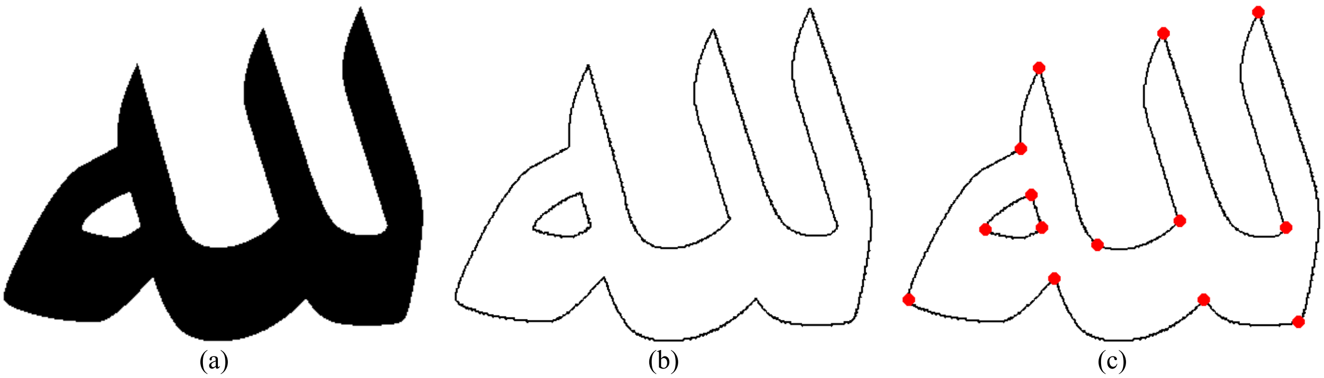


Figure 1. Pre-processing Steps: (a) Original Image, (b) Outline of the image, (c) Corner points achieved..

A. Linear Spline

The curve fitted by a linear spline is a candidate of best fit, but it may not be a desired fit. This leads to the need of introducing some extra treatment in the methodology. This section deals with a form of linear spline. It introduces parameters t's in the description of linear spline defined as follows:

$$P(t) = P_i(1-\theta) + P_{i+1}\theta$$

where

$$\theta\big|_{[t_i, t_{i+1})}(t) = \frac{(t-t_i)}{h_i}, \; h_i = t_{i+1} - t_i .$$

P_i and P_{i+1} are corner or intermediate points of the i^{th} piece.

B. Conic Spline

The curve fitted by a conic spline is a candidate of best fit, but it may not be a desired fit. This leads to the need of introducing some shape parameters in the description of the conic spline. This section deals with a form of conic spline. It introduces shape parameters u's in the description of conic spline defined as follows:

$$P(t) = \frac{P_i(1-\theta)^2 + u_i U_i 2\theta(1-\theta) + P_{i+1}\theta^2}{(1-\theta)^2 + 2u_i\theta(1-\theta) + \theta^2},$$

where

$$U_i = \frac{(V_i + W_i)}{2}, \; V_i = P_i + \frac{h_i D_i}{u_i}, \; W_i = P_{i+1} - \frac{h_i D_{i+1}}{u_i}.$$

D_i and D_{i+1} are the corresponding tangents at corner points P_i and P_{i+1} of the i^{th} piece. The tangent vectors are calculated as follows:

$$\left.\begin{aligned} D_0 &= 2(P_1 - P_0) - \frac{(P_2 - P_0)}{2} \\ D_i &= a_i(P_i - P_{i-1}) + (1-a_i)(P_{i+1} - P_i) \\ D_n &= 2(P_n - P_{n-1}) - \frac{(P_n - P_{n-2})}{2} \end{aligned}\right\},$$

where

$$a_i = \frac{\|P_{i+1} - P_i\|}{\|P_{i+1} - P_i\| + \|P_i - P_{i-1}\|}.$$

Obviously, the parameters u_i's, when equal to 1, provide the special case of quadratic spline. Otherwise, these parameters can be used to loose or tight the curve. This paper proposes an evolutionary technique, namely multilevel coordinate search (MCS), to optimize these parameters so that the curve fitted is optimal. For details of this scheme, the reader is referred to [28].

IV. Proposed Approach for Vectorization

Since, the objective of the paper is to come up with optimal techniques which can provide decent curve fit to the digital data. Therefore, the interest would be to compute the curve in such a way that the sum square error of the computed curve with the actual curve (digitized contour) is minimized. Mathematically, the sum squared distance is given by:

$$S_i = \sum_{j=1}^{m_i}\left[P_i(u_{i,j}) - P_{i,j}\right]^2, \; t_{i,j} \in [t_i, t_{i+1}], \; i = 0,1,...,n-1,$$

where

$$P_{i,j} = (x_{i,j}, y_{i,j}), \quad j = 1,2,\ldots,m_i,$$

are the data points of the ith segment on the digitized contour. The parameterization over t's is in accordance with the chord length parameterization. Thus the curve fitted in this way will be a candidate of best fit. The overall scheme can be explained in an algorithm as follows:

A. Algorithm 1 for Linear Interpolant

The summary of the algorithm, designed for optimal curve design using linear interpolant, is as follows:

Step AG1.1: Input the image.

Step AG1.2: Extract the contours from the image in Step AG1.1.

Step AG1.3: Compute the corner points from the contour points in Step AG1.2 using the method in Section 3.

Step AG1.4: Fit the linear spline curve method of Section 3.A to the corner points achieved in Step AG1.3.

Step AG1.5: IF the curve, achieved in Step AG1.4, is optimal then GO To Step AG1.8, ELSE locate the appropriate intermediate points (points with highest deviation) in the undesired curve pieces.

Step AG2.6: Enhance and order the list of corner and intermediate points achieved in Step AG1.3 and AG1.5.

Step AG2.7: GO TO Step AG1.4.

Step AG2.8: STOP.

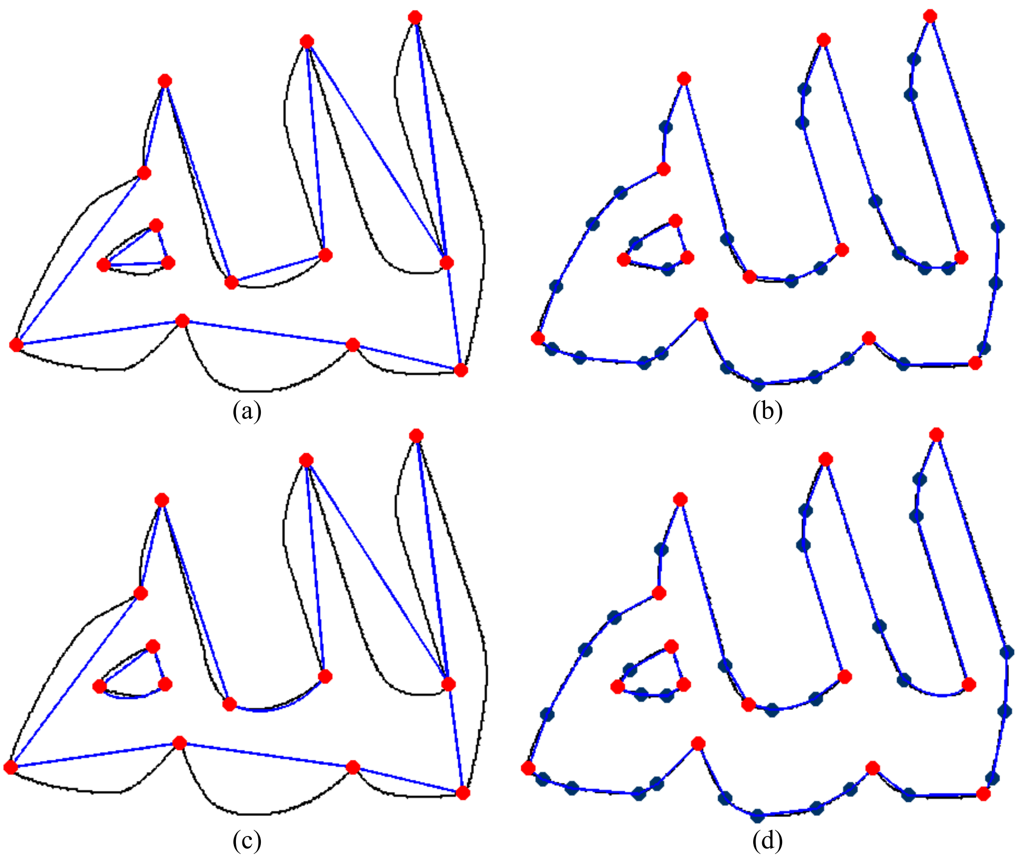


Figure 2. (a) Fitted Outline of the image, (b) Fitted Outline of the image with intermediate points.

B. Algorithm 2 for Conic Interpolant

The summary of the algorithm, designed for optimal curve design using conic interpolant, is as follows:

Step AG2.1: Input the image.

Step AG2.2: Extract the contours from the image in Step AG2.1.

Step AG2.3: Compute the corner points from the contour points in Step AG2.2 using the method in Section 3.

Step AG2.4: Compute the derivative values at the corner / intermediate points.

Step AG2.5: Compute the best optimal values of the shape parameters u_i's using MCS.

Step AG2.6: Fit the spline curve method, of Section 3, to the corner / intermediate points achieved in Step AG2.2.

Step AG2.7: IF the curve, achieved in Step AG2.6, is optimal then GO To Step AG2.10, ELSE locate the appropriate intermediate points (points with highest deviation) in the undesired curve pieces.
Step AG2.8: Enhance and order the list of the corner / intermediate points achieved in Step AG2.3 and AG2.7.
Step AG2.9: GO TO Step AG2.4.
Step AG2.10: STOP.

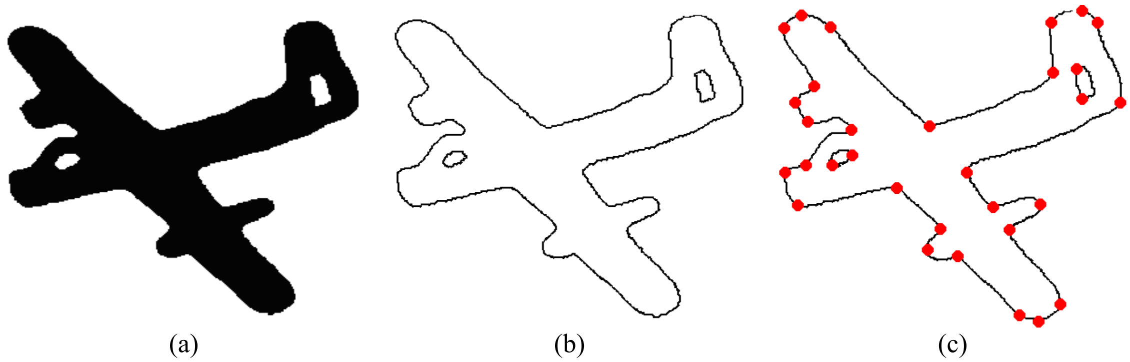


Figure 3. Pre-processing Steps: (a) Original Image, (b) Outline of the image, (c) Corner points achieved, (d) Fitted Outline of the image.

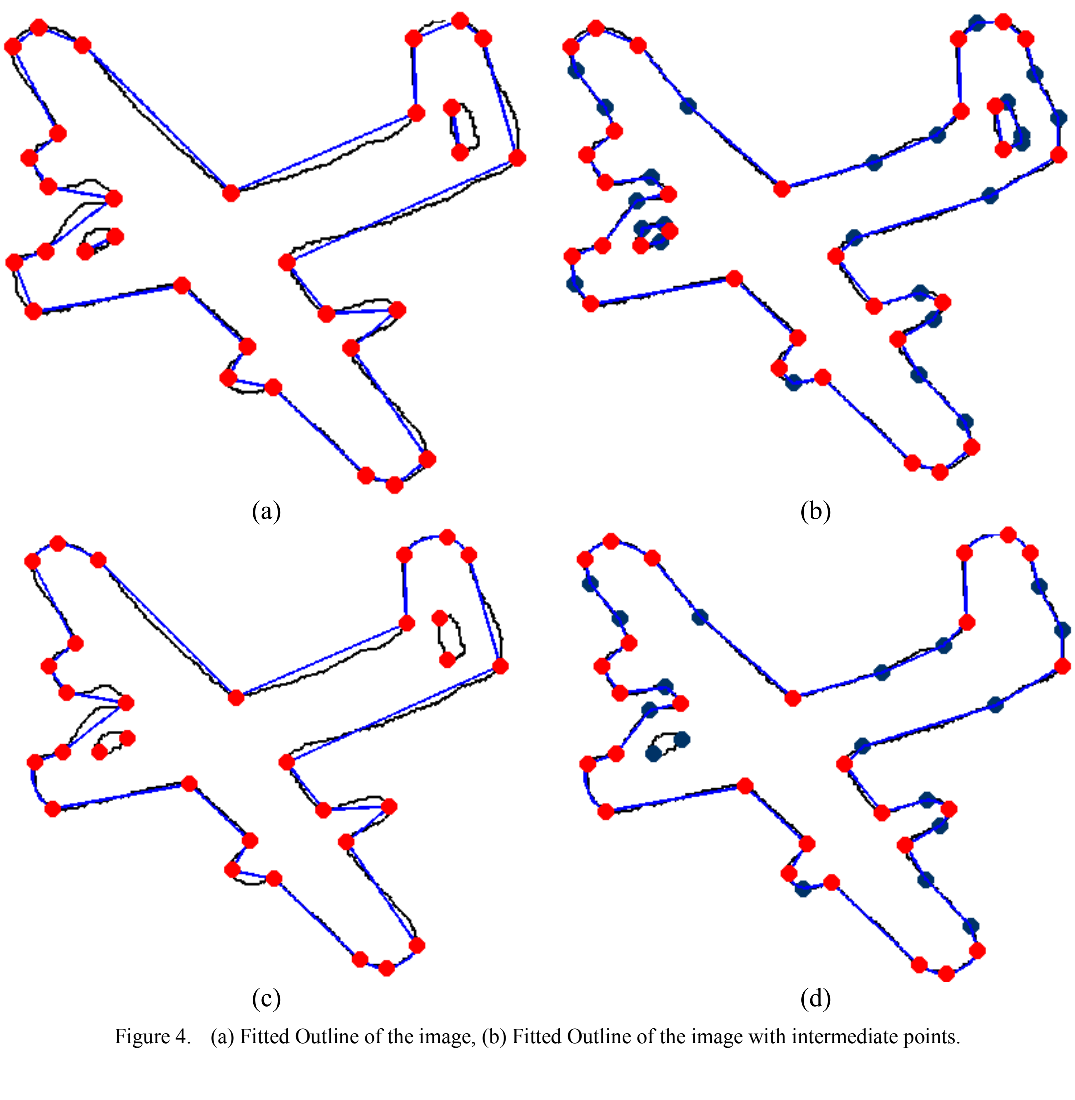


Figure 4. (a) Fitted Outline of the image, (b) Fitted Outline of the image with intermediate points.

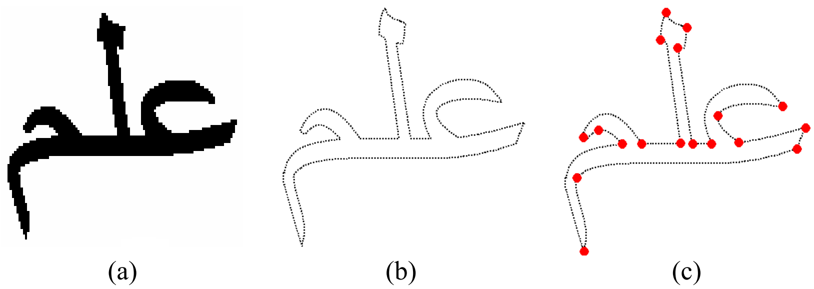


Figure 5. Pre-processing steps for curve fitting (a) Image of a plane, (b) Extracted outline (c) Initial corner points.

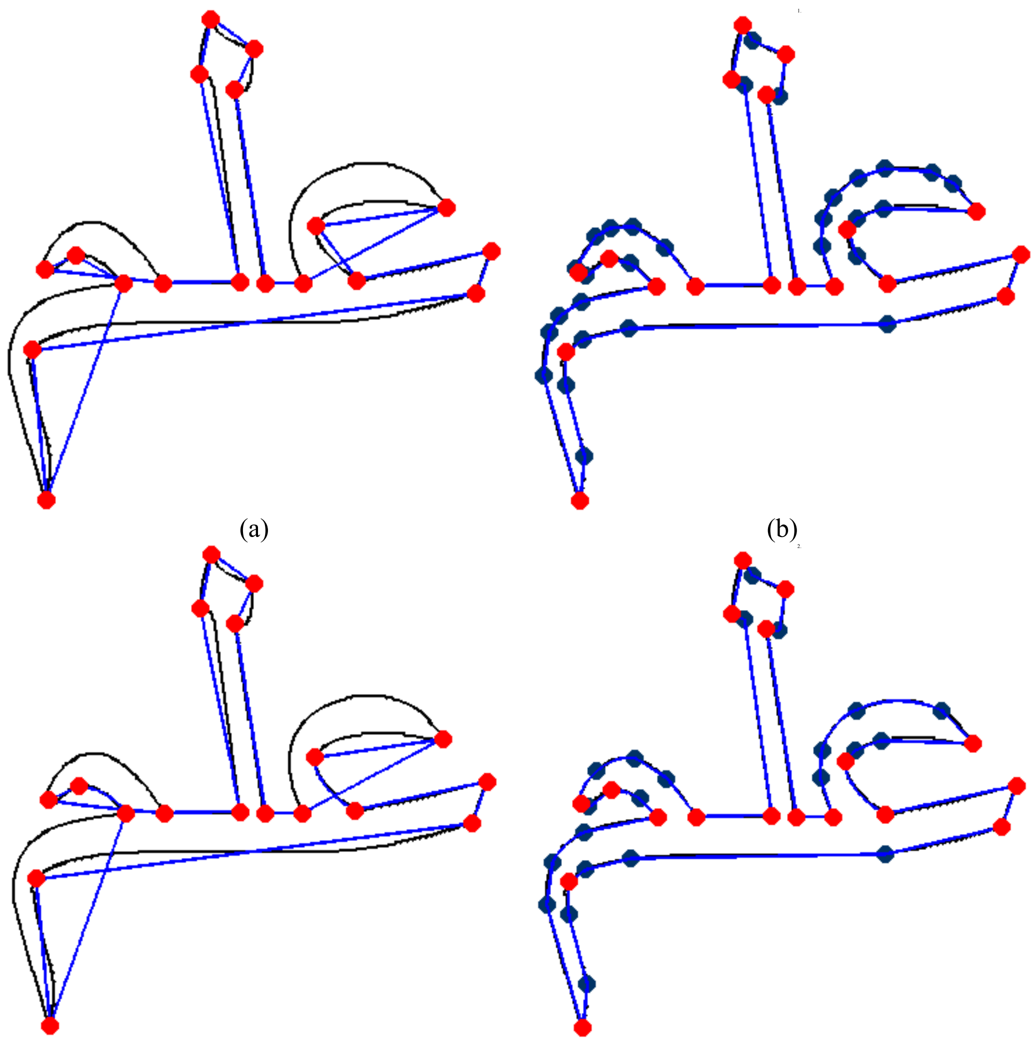


Figure 6. Conic curve fitting (a) without intermediate points (b) with intermediate points.

The above mentioned schemes and the algorithms have been implemented and tested for various images. Reasonably quite elegant results have been observed as can be seen in the following Section of demonstrations.

V. DEMONSTRATIONS

The proposed curve scheme has been implemented successfully in this section. We evaluate the performance of our system by fitting parametric curves to different binary images.

Figure 2 shows the implementation results of the two algorithms for the image in Figure 1(a). Figures 2(a) and 2(b) are the results for the linear scheme, respectively, without and with insertion of intermediate points. Similarly, Figures 2(c) and 2(d) are the results for the conic scheme, respectively, without and with insertion of intermediate points.

Figure 3 shows the implementation results of a "Airplane" image. Figures 3(a), 3(b), 3(c) are respectively the original image of an Airplane, its outline, outline together with the corner points detected. Figure 4 shows the implementation results of the two algorithms for the image in Figure 3(a). Figures 4(a) and 4(b) are the results for the linear scheme, respectively, without and with insertion of intermediate points. Similarly, Figures 4(c) and 4(d) are the results for the conic scheme, respectively, without and with insertion of intermediate points.

TABLE I. NAMES AND CONTOUR DETAILS OF IMAGES

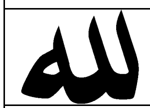
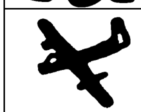
Image	Name	# of Contours	# of Contour Points	# of Initial Corner Points
	Lillah	2	[1522+161]	14
	Plane	3	[1106+61+83]	31
	Ilm	1	[1641]	18

TABLE II. COMPARISON OF NUMBER OF INITIAL CORNER POINTS, INTERMEDIATE POINTS AND TOTAL TIME TAKEN (IN SECONDS) FOR CONIC INTERPOLATION APPROACHES.

Image	# of Intermediate Points in Linear Interpolation	Total Time Taken For Linear Interpolation	
		Without Intermediate Points	With Intermediate Points
Lillah.bmp	29	2.827	3.734
Plane.bmp	24	7.735	8.39
Ilm.bmp	29	4.06	5.032

Figure 5 shows the implementation results of a "Airplane" image. Figures 5(a), 5(b), 5(c) are respectively the original image of an Airplane, its outline, outline together with the corner points detected. Figure 4 shows the implementation results of the two algorithms for the image in Figure 5(a). Figures 6(a) and 6(b) are the results for the linear scheme, respectively, without and with insertion of intermediate points. Similarly, Figures 6(c) and 6(d) are the results for the conic scheme, respectively, without and with insertion of intermediate points.

TABLE III. COMPARISON OF NUMBER OF INITIAL CORNER POINTS, INTERMEDIATE POINTS AND TOTAL TIME TAKEN (IN SECONDS) FOR CONIC INTERPOLATION APPROACHES.

Image	# of Intermediate Points in Conic Interpolation	Total Time Taken For Conic Interpolation	
		Without Intermediate Points	With Intermediate Points
Lillah.bmp	14	19.485	48.438
Plane.bmp	31	20.953	49.356
Ilm.bmp	18	19.594	42.078

One can see that the approximation is not satisfactory when it is achieved over the corner points only. This is specifically due to those segments which are bigger in size and highly curvy in nature. Thus, some more treatment is required for such outlines. This is the reason that the idea to insert some intermediate points is demonstrated in the algorithms. It provides excellent results. The idea of how to insert intermediate points is not explained here due to limitation of space. It will be explained in a subsequent paper.

Tables I to III summarize the experimental results for different bitmap images. These results highlight various information including contour details of images, corner points, intermediate points, total time taken for linear interpolation, and total time taken for conic interpolation.

VI. CONCLUSION AND FUTURE WORK

Two optimization techniques are proposed for the outline capture of planar images. First technique uses simply a linear interpolant and a straight forward method based on distribution of corner and intermediate points. Second d technique uses the multilevel coordinate search to optimize a conic spline to the digital outline of planar images. By starting a search from certain good points (initially detected corner points), an improved convergence result is obtained. The overall technique has various phases including extracting outlines of images, detecting corner points from the detected outline, curve fitting, and addition of extra knot points if needed. The idea of multilevel coordinate search has been used to optimize the shape parameters in the description of a conic spline introduced. The two methods ultimately produce optimal results for the approximate vectorization of the digital contours obtained from the generic shapes. The schemes provide an optimal fit with an efficient computation cost as far as curve fitting is concerned. The proposed algorithms are fully automatic and require no human intervention. The author is also thinking to apply the proposed methodology for another model curve namely cubic. It might improve the approximation process. This work is in progress to be published as a subsequent work.

References

[1] Chetrikov, D., Zsabo, S.: A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves, In Proceedings of the 23rd Workshop of the Australian Pattern Recognition Group, pp. 1751-2184 (1999).

[2] Goshtasby A..A.: Grouping and Parameterizing Irregularly Spaced Points for Curve Fitting, ACM Transactions on Graphics, vol. 19(3), pp. 185-203 (2000).

[3] Reche, P., Urdiales, C., Bandera, A.: Trazegnies, C. and Sandoval, F. , Corner Detection by Means of Contour Local Vectors, Electronic Letters, vol. 38(14) (2002).

[4] Marji, M., Siv, P.: A New Algorithm for Dominant Points Detection and Polygonization of Digital Curves, Pattern Recognition, pp. 2239-2251 (2003).

[5] Wu-Chih Hu, Multiprimitive Segmentation Based on Meaningful Breakpoints for Fitting Digital Planar Curves with Line Segments and Conic Arcs, Image and Vision Computing, pp. 783-789 (2005).

[6] Freeman, H., Davis, L.S.: A corner finding algorithm for chain-coded curves, IEEE Trans. Computers, Vol. 26, pp. 297-303 (1977).

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

[8] Sarfraz, M.: Representing Shapes by Fitting Data using an Evolutionary Approach, International Journal of Computer-Aided Design & Applications, vol. 1(1-4), pp 179-186 (2004).

[9] Sarfraz, M. and Khan, M.A.: An Automatic Algorithm for Approximating Boundary of Bitmap Characters, Future Generation Computer Systems, pp. 1327-1336 (2004).

[10] Sarfraz, M.: Some Algorithms for Curve Design and Automatic Outline Capturing of Images, International Journal of Image and Graphics, pp. 301-324 (2004).

[11] Hou, Z. J., Wei, G.W.: A New Approach to Edge Detection, Pattern Recognition, pp. 1559-1570 (2002).

[12] Sarfraz, M.: Computer-Aided Reverse Engineering using Simulated Evolution on NURBS, International Journal of Virtual & Physical Prototyping, Taylor & Francis, vol. 1(4), pp. 243 – 257 (2006).

[13] Sarfraz, M., Riyazuddin, M., Baig, M.H.: Capturing Planar Shapes by Approximating their Outlines, International Journal of Computational and Applied Mathematics, Elsevier Science, vol. 189(1-2), pp. 494 – 512 (2006).

[14] Sarfraz, M., Rasheed, A.: A Randomized Knot Insertion Algorithm for Outline Capture of Planar Images using Cubic Spline, The Proceedings of The 22th ACM Symposium on Applied Computing (ACM SAC-07), Seoul, Korea, pp. 71 – 75, ACM Press (2007).

[15] Kano, H., Nakata, H., Martin, C.F.: Optimal Curve Fitting and Smoothing using Normalized Uniform B-Splines: A Tool for Studying Complex Systems, Applied Mathematics and Computation, pp. 96-128 (2005).

[16] Yang, Z., Deng, J., Chen, F.: Fitting Unorganized Point Clouds with Active Implicit B-Spline Curves, Visual Computer, pp. 831-839 (2005).

[17] Lavoue, G., Dupont, F., Baskurt, A.: A New Subdivision Based Approach for Piecewise Smooth Approximation of 3D Polygonal Curves, Pattern Recognition, vol. 38, pp. 1139-1151 (2005).

[18] Yang, H., Wang, W., and Sun, J.: Control Point Adjustment for B-Spline Curve Approximation, Computer Aided Design, vol. 36, pp. 639-652 (2004).

[19] Horng, J. H.: An Adaptive Smoothing Approach for Fitting Digital Planar Curves with Line Segments and Circular Arcs, Pattern Recognition Letters, vol. 24(1-3), pp. 565-577 (2003).

[20] Sarkar, B., Singh, L.K., Sarkar, D.: Approximation of Digital Curves with Line Segments and Circular Arcs using Genetic Algorithms, Pattern Recognition Letters, vol. 24, pp. 2585-2595 (2003).

[21] Yang, X.: Curve Fitting and Fairing using Conic Spines, Computer Aided Design, vol. 6(5), pp. 461-472 (2004).

[22] Yang, X.N., Wang G.Z.: Planar Point Set Fairing and Fitting by Arc Splines, Computer Aided Design, pp. 35-43 (2001).

[23] Sarfraz, M.: Designing Objects with a Spline, International Journal of Computer Mathematics, Taylor & Francis, vol. 85(7) (2008).

[24] Huyer, W., Neumaier, A.: Global Optimization by Multilevel Coordinate Search, Journal of Global Optimization, vol. 14, pp. 331–355 (1999).

[25] Multi-level Coordinate Search, http://www.mat.univie.ac.at/~neum/software/mcs/.

[26] Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB, 2nd Ed., Gatesmark Publishing (2009).

[27] Nixon, M.S., Aguado, A.S.: Feature extraction and image processing, Elsevier (2008).

[28] Sarfraz, M. (2009), Outline Capture of Images by Multilevel Coordinate Search on Cubic Splines, Lecture Notes in Artificial Intelligence, A. Nicholson and X. Li (Eds.): LNAI 5866, Springer-Verlag Berlin Heidelberg, pp. 636–645.