Proceedings of 2012 4th International Conference on Machine Learning and Computing
IPCSIT vol. 25 (2012) © (2012) IACSIT Press, Singapore

Vectorizing Image Outlines using Spline Computing Approach

Muhammad Sarfraz[+]

Department of Information Science, Kuwait University, Adailiya Campus, P.O. Box 5969, Safat 13060, Kuwait

Abstract. Linear and conic vectorization techniques have been reviewed and compared for the outline capture of the planar images. The idea of simulated annealing has been incorporated to optimize the shape parameters in the description of the generalized conic spline. However, a straightforward approach has been used for linear spline case because of having no degree of freedom. The two methods ultimately produce optimal results for the approximate vectorization of the digital contours obtained from the generic shapes. Demonstrations and a comparative study also make the essential parts of the paper.

Keywords: Vectorization, corner points, generic shapes, curve fitting, spline

1. 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 modelling [1] is one of the important phases of capturing and vectorizing outlines of images. It plays a 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 [3-6], 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, review and comparative study of two approaches using linear and conic interpolations. The linear interpolant approach is straight forward. However, the conic approach [6] is inspired by an optimization algorithm based on simulated annealing (SA) by Kirkpatrick et. al. [8]. Tis paper motivates the author to make a comparative study of conic technique versus the liner spline optimization technique 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 objectives. The algorithm comprises of various phases to achieve the target. First of all, it finds the contour [5] of the gray scaled bitmap images. Secondly, it uses the idea of corner points [2] to detect corners. That is, it detects the corner points on the digital contour of the generic shape under consideration. These phases are considered as preprocessing steps. Linear and conic interpolants are then used to vectorize the outline. The idea of simulated annealing (SA) is used to fit a conic spline [6] 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. In case of poor approximation, the insertions of intermediate points are made as long as the desired approximation or fit is achieved.

[+] Corresponding author. Tel.: +965 2 498 3109; fax: +(please specify).
E-mail address: prof.m.sarfraz@gmail.com.

The organization of the paper is as follows, Section 2 discusses about preprocessing steps which include finding the boundary of planar objects and detection of corner points. Section 3 is about the interpolant forms of linear and conic spline curves. Overall methodology of curve fitting is reviewed in Section 4, it includes the idea of knot insertion as well as the algorithm design for the proposed vectorization schemes. Demonstration and comparative study of the two schemes is presented in Section 5. Finally, the paper is concluded in Section 6.

2. Preprocessing

The proposed schemes start with finding the boundary of the generic shape and then using the output to find the corner points. The image of the generic shapes 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 [7], 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).

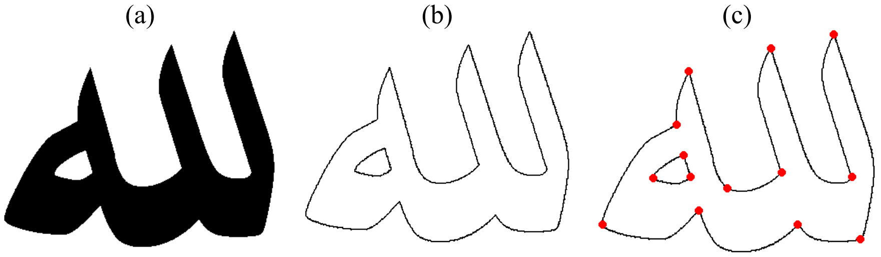


Fig. 1: Pre-processing Steps: (a) Original Image, (b) Outline of the image, (c) Corner points achieved.

3. 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 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, $i = 0,1,......,n$.

3.1. 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:

$$P(t) = P_i(1-\theta) + P_{i+1}\theta \text{ where } \theta\big|_{[t_i,t_{i+1})}(t) = (t-t_i)/h_i,\ h_i = t_{i+1} - t_i.$$

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

3.2. 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:

$$P(t) = \frac{P_i(1-\theta)^2 + u_i U_i 2\theta(1-\theta) + P_{i+1}\theta^2}{(1-\theta)^2 + 2u_i\theta(1-\theta) + \theta^2}, \text{ where } U_i = \frac{(V_i + W_i)}{2},\ V_i = P_i + \frac{h_i D_i}{u_i},\ W_i = P_{i+1} - \frac{h_i D_{i+1}}{u_i}.$$

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:

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

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 tighten the curve. This paper proposes an evolutionary technique, namely simulated annealing (SA), to optimize these parameters so that the curve fitted is optimal. For the details of SA approach, the reader is referred to [9].

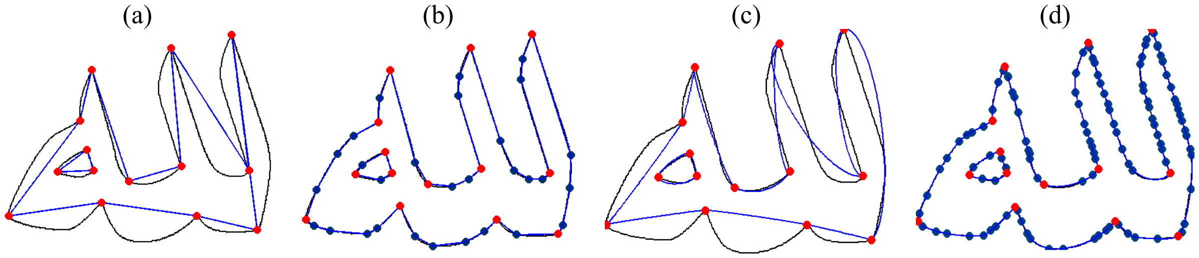


Fig. 2: Results of linear scheme : (a) Fitted Outline of the image, (b) Fitted Outline of the image with intermediate points. Results of conic scheme: (c) Fitted Outline of the image, (d) Fitted Outline of the image with intermediate points.

4. Curve Fitting and Spline

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:

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

where $P_{i,j} = (x_{i,j}, y_{i,j})$, $j = 1,2,\dots,m_i$, 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:

4.1. Algorithm 1 for Linear Interpolant

The summary of the algorithm, designed for optimal curve 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.

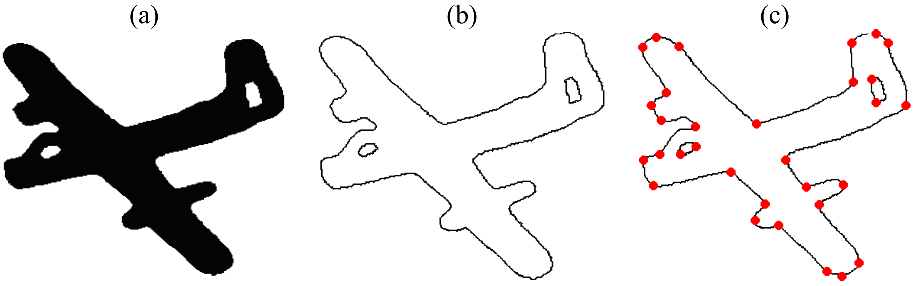


Fig. 3: Pre-processing Steps: (a) Original Image, (b) Outline of the image, (c) Corner points achieved, (d) Fitted Outline of the image.

4.2. 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 SA.
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.

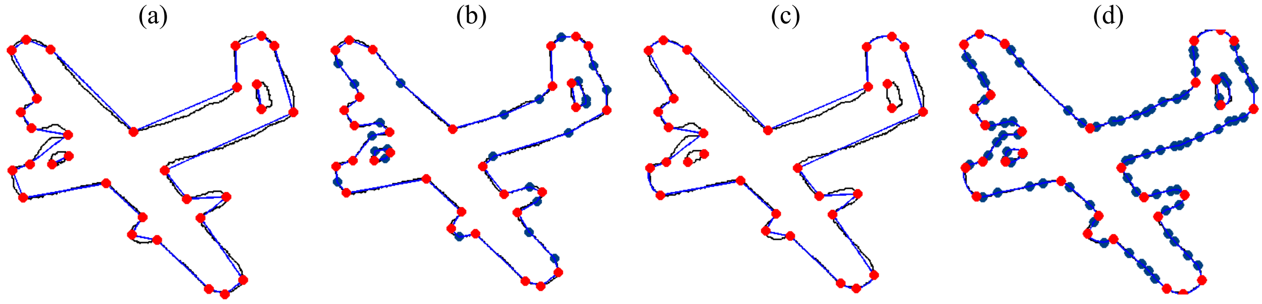


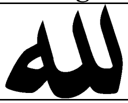
Fig. 4: Results of curve fitting. Linear curve fitting : (a) Fitted Outline of the image, (b) Fitted Outline of the image with intermediate points. Conic curve fitting: (c) Fitted Outline of the image, (d) Fitted Outline of the image with intermediate points.

5. Demonstrations

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

Figure 2 shows the implementation results of the two algorithms for the image "Lillah" 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.

Table 1: 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

Figures 3 and 4 show the implementation results of a "Plane" image. Figures 3(a), 3(b), 3(c) are respectively the original image of the Plane, its outline, outline together with the corner points detected. Figure 4 shows the implementation results of the two algorithms for the "Plane" 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 2: 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

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, II and 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.

Table 3: 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

6. Concluding Remarks

Two optimization techniques are reviewed and copmaraed 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 technique uses the simulated annealing to optimize a conic spline to the digital outline of planar images. The idea of simulated annealing has been used to optimize the shape parameters in the description of a conic spline introduced. The comparative study shows that 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.

7. Acknowledgements

This work was supported by Kuwait University, Research Grant No. [WI 05/10].

8. References

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

[2] 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).

[3] 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).

[4] 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).

[5] Sarfraz, M. and Sarfraz, M.I. (2009), Capturing Image Outlines using Spline Computing Approach, The Proceedings of The 5th International Conference on Signal-Image Technology & Internet–Based Systems (SITIS-2009), November 30 – December 03, 2009, Marrakech, Morocco , pp. 126 - 132, IEEE Computer Society Press.

[6] Sarfraz, M. (2011), Capturing Image Outlines using Simulated Annealing Approach with Conic Splines, The Proceedings of the International Conference on Information and Intelligent Computing (ICIIC 2011), Hong Kong, China, November 25-27, 2011, pp. 152 – 157, IPCSIT vol. 18, IACSIT Press, Singapore.

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

[8] Kirkpatrick, S., C. D. Gelatt Jr., M. P. Vecchi, Optimization by Simulated Annealing, *Science*, Vol. 220(4598), pp. 671-680, 1983.

[9] Sarfraz, M. (2010), Vectorizing Outlines of Generic Shapes by Cubic Spline using Simulated Annealing, International Journal of Computer Mathematics, Taylor & Francis, Vol. 87(8), pp. 1736 – 1751.