2011 International Conference on Information and Intelligent Computing
IPCSIT vol.18 (2011) © (2011) IACSIT Press, Singapore

Capturing Image Outlines using Simulated Annealing Approach with Conic Splines

Muhammad 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. The idea of Simulated Annealing has been used to optimize the shape parameters in the description of the generalized conic spline introduced.

Keywords: Imaging; optimization; Simulated Annealing; generic shapes; curve fitting

1. Introduction

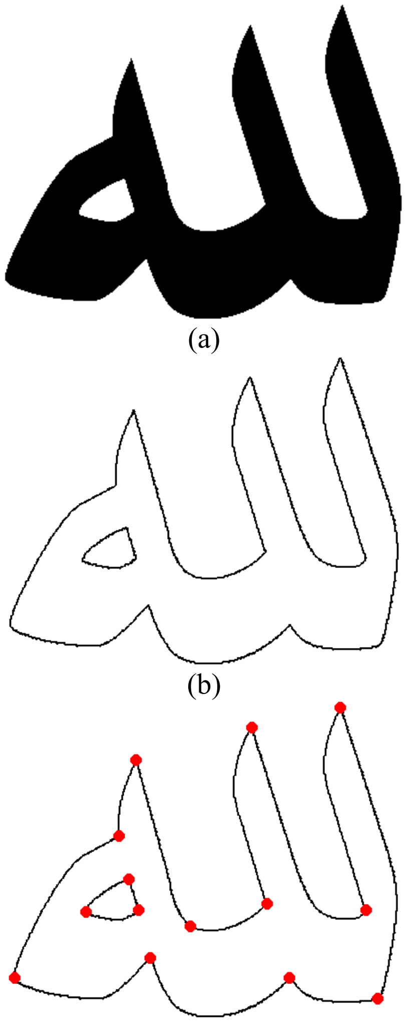
Capturing outlines of images is one of the important problems of computer graphics, vision, and imaging. Various mathematical and computational phases are involved in the whole process. Curve modeling [4-6] plays significant role in it. The representation of planar objects in terms of curves has many 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 [2-3], it is still desired to proceed further to explore more advanced and interactive strategies.

This work is inspired by an optimization algorithm based on simulated annealing (SA) by Kirkpatrick et. al. [9]. It motivates the author to introduce a robust technique for the outline capture of generic shapes. The algorithm comprises of various phases to achieve the target. First of all, it finds the contour of the gray scaled bitmap image [7-8]. Secondly, it uses the idea of corner points [1] 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. The idea of Simulated Annealing (SA) is then 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.

2. Preprocessing

The proposed scheme starts with first finding the boundary of the generic shape and then using the output to find the corner points. Chain codes [8], in this paper, have been used for finding the boundary. Demonstration of the method can be seen in Figure 1(b) which is the contour of the bitmap image shown in Figure 1(a).

Corner points 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).



(a)

(b)

(c)

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

3. Curve Fitting and Spline

The curve fitted, to the corner points, 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

$$\theta\big|_{[t_i,t_{i+1})}(t) = \frac{(t-t_i)}{h_i},\ h_i = t_{i+1} - t_i,\ U_i = \frac{(V_i + W_i)}{2},$$

and

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

F_i and F_{i+1} are corner points of the i^{th} piece. D_i and D_{i+1} are the corresponding tangents at corner points.

Obviously, the parameters u_i's, when equal to 1, provide the special case of quadraticic spline. Otherwise, these parameters can be used to loose or tight the curve. This paper proposes an evolutionary technique, namely Simulated Annealing (SA), to optimize these parameters so that the curve fitted is optimal.

To construct the parametric conic 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, $i = 0,1,......,n$. The tangent vectors D_i's can be calculated using some appropriate method like arithmetic means.

Since, the objective of the paper is to come up with an optimal technique which can provide a 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.

4. Simulated Annealing (SA)

Simulated annealing (SA) is an optimization technique [9]. Our interest is to optimize the values of curve parameters u such that the defined curve fits as close to the original contour segments as possible. Note that we apply SA independently for each segment of a contour that we have identified using corner points. SA is applied sequentially on each of the segments, generating an optimized fitted curve for each segment. The algorithm is run until the maximum allowed time is reached, or an optimal curve fitting is attained. Derivation of the SA algorithm and underlying theory can be found in [9].

5. Proposed Approach

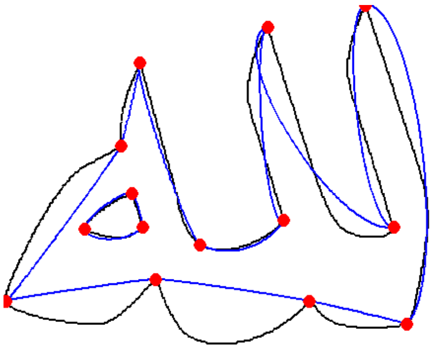
Once we have the bitmap image of a generic shape, the boundary of the image can be extracted using the method described in Section 2. After the boundary points of the image are found, the next step is to detect corner points as explained in Section 2. This corner detection technique assigns a measure of 'corner strength' to each of the points on the boundary of the image. This step helps to divide the boundary of the image into n segments. Each of these segments is then approximated by interpolating spline described in Section 3. The initial solution of spline parameters (u) are randomly selected within the range [-1, 1].

After an initial approximation for the segment is obtained, better approximations are obtained through SA to reach the optimal solution. We experiment with our system by approximating each segment of the boundary using the generalized conic spline of Section 3. Each boundary segment is approximated by the spline. The shape parameters (u) in the conic spline provide greater flexibility over the shape of the curve. These parameters are adjusted using SA to get the optimal fit.

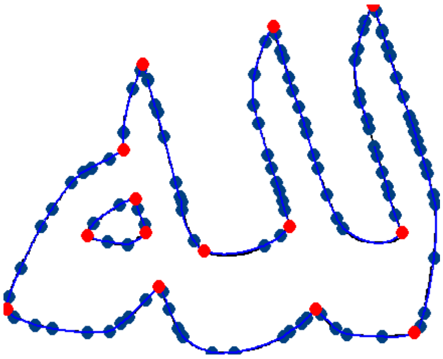
For some segments, the best fit obtained through iterative improvement may not be satisfactory. In that case, we subdivide the segment into smaller segments at points where the distance between the boundary and parametric curve exceeds some predefined threshold. Such points are termed as intermediate points. A new parametric curve is fitted for each new segment.

6. Demonstration

The methodology, in Section 5, has been implemented practically and the proposed curve scheme has been implemented successfully. We evaluate the performance of the system by fitting parametric curves to different binary images.



(a)

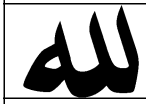


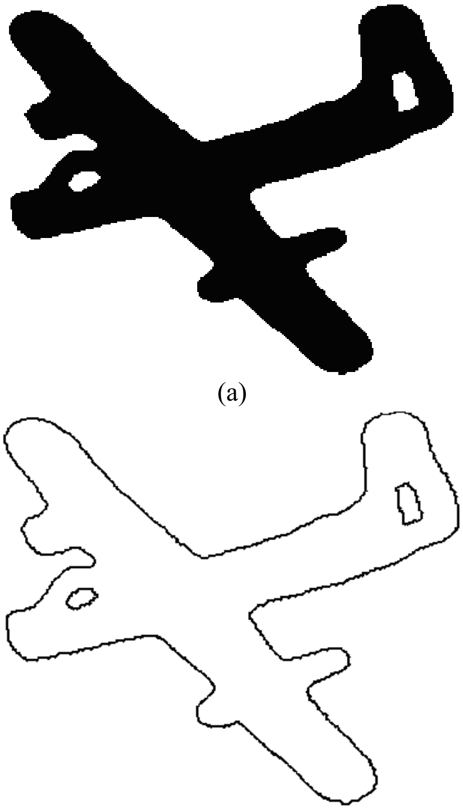
(b)

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

Fig. 2(a) shows the implementation results of the algorithm with SA for the original image of an Arabic language word "Lillah" in Fig. 1. One can see that the approximation is not satisfactory, 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. One of the idea is to insert some intermediate points, this is demonstrated in Fig. 2(b) where excellent result has been achieved. 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.

TABLE I. NAMES AND CONTOUR DETAILS OF IMAGES

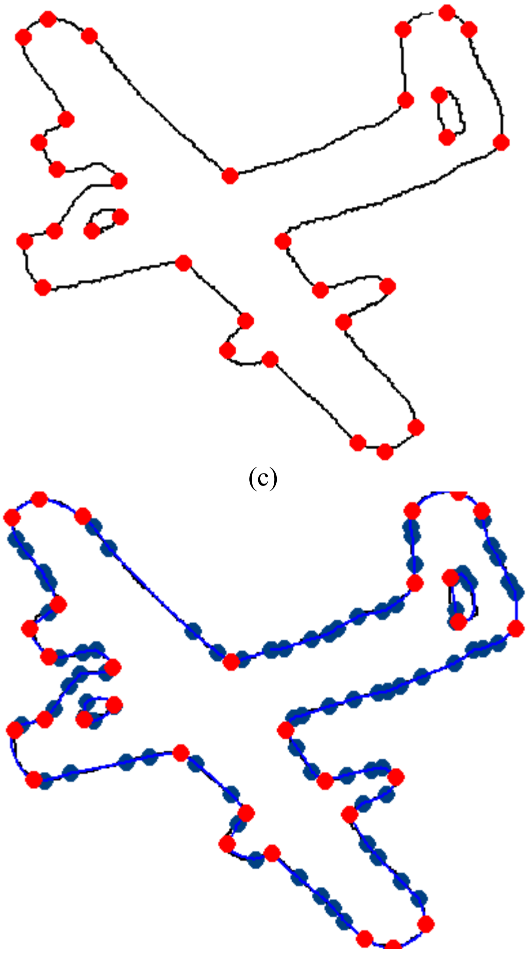
Image	Name	# of Contours	# of Contour Points
	Lillah	2	[1522+161]
	Plane	3	[1106+61+83]



(a)

(b)

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



(c)

(d)

Figure 4. Pre-processing Steps: (a) Original Image, (b) Outline of the image, (c) Corner points achieved, (d) Fitted Outline of the image.

Figure 3(a) shows another image of an aeroplane. Its outline is shown in Figure 3(b). Figure 4(a) demonstrates the corner detection of Figure 3(b). The implementation result of the algorithm with SA can be see in Figure 4(b), this is the fitted outline at the final iteration.

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

Image	**# of Initial Corner Points**	**# of Intermediate Points in Conic Interpolation**	**Total Time Taken For Conic Interpolation**	
			Without Intermediate Points	**With Intermediate Points**
Lillah.bmp	14	108	85.735	660.238
Plane.bmp	31	79	78.175	368.242

Tables I to II summarize the experimental results for different bitmap images. These results highlight various information including contour details of images (Table I) and intermediate points (Table II).

7. Conclusion and Future Work

A global optimization technique, based on Simulated Annealing, is proposed for the outline capture of planar images. The proposed technique uses the Simulated Annealing 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 Simulated Annealing has been used to optimize the shape parameters in the description of a conic spline introduced. The spline method ultimately produces optimal results for the approximate vectorization of the digital contour obtained from the generic shapes. It provides an optimal fit with an efficient computation cost as far as curve fitting is concerned. The proposed algorithm is fully automatic and requires 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.

8. Acknowledgment

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

9. 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] 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).

[3] 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).

[4] 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).

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

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

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

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

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