

# Role of Corner Detection in Capturing Shape Outlines

Asif Masood<sup>1</sup> Shaiq A. Haq<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, University of Engineering and Technology, Lahore, Pakistan.

## Abstract

In this paper, a role of corner detection in outline capturing system is studied/discussed, which is normally used as phase 1 of capturing. Three outline capturing techniques are studied, one without involving any corner detection and two techniques with most suited corner detectors. One common curve approximation method, with cubic Bezier curves, is used in all above techniques in order to study how a corner detector can affect the overall capturing system. Various parameters are introduced for measuring the quality of captured outlines. It is concluded in the end that detected of corner points plays an important role in any outline capturing system.

**Keywords:** Outline capturing; Curve approximation; Cubic Bezier curves; Corner detection; Data points

## 1. Introduction

Shape of an object under image processing can be represented by its boundary points [6] or the interior of a shape [7]. Boundary or outline representation is more efficient, as it takes less memory and preserves the complete shape of an object. This representation also proves heavy in various image processing applications (like processing videos), which lead to outline capturing systems [12]. In that, outlines are represented with few data points. Many researchers have made efforts [11,12] to capture various 2D shapes with minimum data points and approximation error. Outline capturing is normally a two step process. First, detection of initial data points using some corner detector. Second, finding other needed data points using curve fitting or curve approximation techniques.

Generally corners are the intersection points of straight lines in a polygon. Substitute to the corners in a curve, are the high curvature points. Attneave [1] proposed that information along a contour is concentrated in the regions of high magnitude of curvature. Precise detection of corner points can play very important role in capturing systems. In this paper, we show how the detected corners can effect the over all shape of captured outlines, number of data points and approximation error.

Various curve modals are used [11,12] for second phase of capturing outlines. We use cubic Bezier curves for its efficient implementation and design flexibility. Effects of changing corner detection algorithm are studied using the same curve approximation technique. Various quality measuring parameters are defined and used for accessing the role of corner detection in capturing outlines. A curve approximation technique with cubic Bezier curves is presented in section 2. Three outline capturing methodologies are introduced in section 3. Capturing results are demonstrated/ compared in section 4 and section 5 concludes this presentation.

## 2. Curve Approximation

In curve approximation, an approximating spline curve is determined for a given curve. Cubic Bezier curves are used as approximating curves in this paper. These are generated with four control points  $P_i = (x_i, y_i)$ , with  $i$  varying from 0 to 3. Data points of approximating cubic Bezier curve are its control points, which are to be determined. Control points  $P_0$  &  $P_3$  are the two endpoints of given curve and the control points  $P_1$  &  $P_2$  are the two approximating points which lay along the tangents of curve ( $T_1$  &  $T_2$ ) at its two endpoints  $P_0$  &  $P_3$  respectively.

Search for the control points  $P_1$  &  $P_2$  starts from the control points  $P_0$  &  $P_3$  along the tangents  $T_1$  &  $T_2$  respectively. Control points  $P_1$  &  $P_2$  keep moving till approximation error ( $AE$ ) is minimized. In this algorithm (fig. 1),  $AE$  is the area between two curves and  $M_1$  &  $M_2$  represents one pixel movement of  $P_1$  &  $P_2$  in the direction of their respective tangents  $T_1$  &  $T_2$ .

1.  $P_1 = P_0$
2.  $P_2 = P_3$
3. Calculate  $M_1$  and  $M_2$
4. Calculate  $AE$
5. Do  $P_1 = P_1 \pm M_1$  While( $AE$  reduces)
6. Do  $P_2 = P_2 \pm M_2$  While( $AE$  reduces)
7. Repeat step 5 and 6 till  $P_1$  or  $P_2$  does not change

Fig. 1: Control points search algorithm.

Given curve is subdivided into two sub-segments after curve approximation if the closest distance

between two curves, at any point along the curve, goes beyond specified threshold error limit. The curve is divided from the maximum error point. New sub-segments are then processed for curve approximation again. Subdivision of curve segments will continue recursively till approximation error for each curve segment is within given threshold limits. Acceptable error limit depends upon the size and resolution of given curve. We use two pixel distance as error threshold in our method.

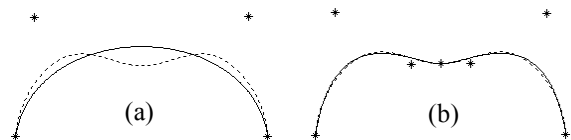


Fig. 2: Curve approximation result. (a) Before segment subdivision. (b) After segment subdivision.

### 3. Outline Capturing

Generally outline capturing is a two step process involving corner detection and curve approximation. The curve approximation technique discussed above can be used for approximation of any given curves. In that, segment subdivision ensures that the approximated curve remains under threshold error limits. Therefore this technique should be able to capture the complete outline without involving any corner detection. Cubic Bezier approximation can be applied by taking the outline as one curve segment, which would be subdivided into sub-segments during recursive segment subdivision.

An outline, *img1* in fig. 3a, was tested with this method. The resultant captured outline along with detected data points is shown in fig. 3b. Captured outline was drawn over original in fig. 3c. Subdivision points are marked with asterisks which are the control points  $P_0$  &  $P_3$  and dots indicate the control points  $P_1$  &  $P_2$ . The captured outline is shown in fig. 3d. Fig 3e is the final end product in the

form of data points i.e., cubic Bezier control points of approximating outline.

Although the outline capturing results presented in fig 3 looks quite acceptable but has some drawbacks as well. We stated curve approximation from a very large segment for example outer loop of *img1*, which was recursively distributed into 9 sub-segments during the capturing process. This involved calculation of approximation error and new approximating cubic Bezier curve after each new segment subdivision, which makes this method computationally very expensive.

### 3.1. Capturing with IPAN99

Outline capturing system can improve by using some corner detector due to its inherent properties [3], which a simple recursive segment division process cannot exhibit. These properties may include robustness to noise, detection of precise corner position, single response (each corner should be detected only once), corner selectivity (rate of correct detections should be high and the wrong ones should be low) and computational efficiency. Readers are referred to some commonly used corner detectors [2-4,9,10]. Sarfraz and Khan [12] used a corner detector by Quddus [8] for capturing outlines of Arabic fonts. Corner detector by Chetverikov and Szabo [3] is most frequently used for outline capturing [11] which is referred as IPAN99 in this paper.

We processed the outline in fig. 3a (*img1*) for demonstration of capturing result with this corner detector as well. The captured outline over original is shown in fig. 4a. Detected corners and subdivision points are shown with circles and asterisks respectively. Other data points (control points  $P_1$  &  $P_2$ ) are omitted for simplicity. Although the number of detected data points do not reduce (for this outline) by using this corner detector but the overall shape of captured outline (fig. 4b) does improve and also there is considerable reduction in computation time (table 1).

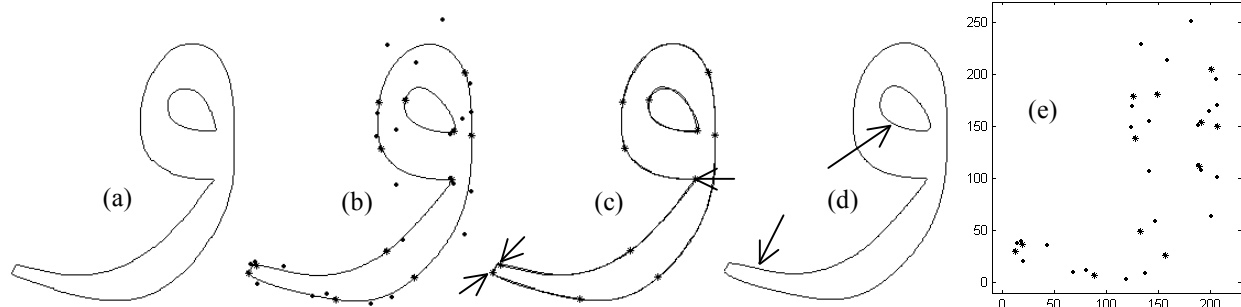


Fig. 3: Outline capturing of *img1* without corner detection. (a) Original outline. (b) Captured outline with data points. (c) Two outline drawn over each other. Asterisks indicate segment subdivision points. (d) Captured outline. (e) Set of detected data points (cubic Bezier control points).



Fig. 4: Outline capturing with corner detector IPAN99. (a) Captured outline over original (img1). Corners and subdivision points are marked with circles and asterisks respectively. (b) Captured outline.

### 3.2. Capturing with SAM04

Overall performance of capturing system did improve by using a corner detector IPAN99. We can observe from the capturing results in fig. 4 that the central position of two closely located corners was picked up as a single corner (indicated with arrow in fig 4a), which resulted in deterioration of the shape of captured outline (indicated with arrow in fig 4b). IPAN99 may have its own advantages but we observe another corner detector [10], most suitable for using in outline capturing systems. It will be referred as SAM04 in this paper. In addition to other advantages as a corner detector, it improves capturing results. It provides sharpness of each detected corner which is further used for outline capturing, hence reducing computation time also. Capturing results, using this corner detector, for the same outline (img1) are shown in fig. 5. One can observe reduction in data points and overall approximation error. Quantitative comparisons of these results are discussed in section 4.

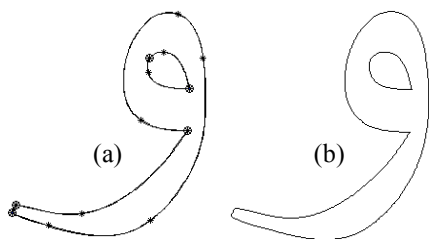


Fig. 5: Outline capturing with corner detector SAM04. (a) Captured outline over original (img1). Corners and subdivision points are marked with circles and asterisks respectively. (b) Captured outline.

## 4. Result Comparison

Outline capturing system consist of two phases i.e., corner detection and curve approximation. Both phases play an important role in outline capturing. In this paper, role of corner detector is studied without

changing the curve approximation technique. For this purpose, three methods were discussed in section 3 for obtaining results. Test results of four outline shapes (img1, img2 & img3) are used for demonstration/ comparison of capturing results. The quantitative comparisons are made on the basis of number of segments, number of data points, maximum error (Max Err), average error (Avg Err) and computation time (Comp Time).

The comparative results for all the test outlines are given in table 1. For img1 in fig 3, 4 & 5, outline capturing with IPAN99, as compared with no corner detector, does not cause any reduction in segments/data points, however there is an improvement in Max Err, Avg Err and Comp Time. Generally, improvement in certain result cost degradation of some other parameter(s) but outline capturing with SAM04, displayed considerable improvement is all computed parameters.

For img2 in fig 6, the captured outline without corner detection consists of 64 data points having computation time of 7.91 secs. With IPAN99, data points and computation time reduced to 52 and 2.78 respectively. With this improvement slight increase in Max Err value is quite acceptable. Its Avg Err value has actually decreased which is the better representation of overall approximation error. With SAM04, data points and computation time further reduced to 46 and 2.49. Max Err value is same for both corner detectors but there is reduction in Avg Err.

For img3, there is improvement of Max Err and Avg Err value with IPAN99 and major improvement in Comp Time. Also note in capturing without corner detection, one corner of outline (indicated with arrow) was not approximated well. With SAM04 compared to IPAN99, all parameters improved except Max Err which remained equal.

## 5. Conclusion

Role of corner detection algorithm in an outline capturing technique is highlighted in this paper. Capturing results are obtained by using a common outline capturing with varying corner detectors. First, capturing without any corner detection is performed, second with corner detector IPAN99 and third with SAM04. Their results are tested/compared under various quality measuring parameters. It can be concluded that proper corner detection before actual curve approximation improves the overall results. Better corner detectors are expected to improve the quality of captured outlines.



Fig 6: Capturing results of img2 & img3 are shown in row 1 & 2 respectively. Column 1 shows the original outline and column 2, 3 & 4 shows captured outlines without corner detection, with IPAN99 and with SAM04 respectively. The captured outlines are drawn over original for easy comparison. Circles and asterisks are the detected corner and subdivision points respectively.

Shape	Corner Detector	Seg	Data Pts	Max Err	Avg Err	Comp Time
Img1	None	13	40	1.78	.60	6.738
	IPAN99	13	40	1.69	.48	4.657
	SAM04	12	37	1.31	.46	4.219
Img2	None	21	64	1.84	.49	7.907
	IPAN99	17	52	1.92	.48	2.769
	SAM04	15	46	1.92	.47	2.484
Img3	None	24	73	1.92	.55	9.875
	IPAN99	24	73	1.85	.52	5.985
	SAM04	22	67	1.85	.51	5.718

Table 1: Quantitative results of outline capturing with different corner detectors.

## References

- [1] F. Attneave, "Some informational aspects of visual perception", *Psychological Review*, vol. 61, pp. 183–193, 1954.
- [2] H.L. Beus, S.S.H. Tiu, "An Improved Corner Detection Algorithm based on Chain Coded Plane Curves", *Pattern Recognition*, 20:291-296, 1987.
- [3] D. Chetverikov, Z. Szabo, "A Simple and Efficient Algorithm for Detection of High Curvature Points in Planner Curves", *Proc. of 23rd workshop of Australian Pattern Recognition Group, Steyr*, pp.175-184, 1999.
- [4] H. Freeman, L.S. Davis, (1977), "A Corner Finding Algorithm for Chain-Coded Curves", *IEEE Trans. Computers*, 26:297-303, 1977.
- [5] D. Hearn, M.P. Baker. *Computer Graphics*. Prentice Hall publication, 1997.
- [6] B. Kartikeyan and R Dhellapa, "Stochastic models for closed boundary analysis: representation and reconstruction", *IEEE trans. on info. theory*, vol. 27, pp. 627-637, 1981.
- [7] J.J. Koenderink and A.J.V. Doorn, "The internal representation of solid shape with respect to vision", *Biol. Cyber.* vol. 32, pp. 211-216, 1979.
- [8] A. Quddus, M.M. Fahmy, "A fast Wavelet based corner detection technique", *Electronic Letters*, vol 35(4), pp. 287-288, 1999.
- [9] A. Rosenfeld and J.S. Weszka. "An improved method of angle detection on digital curves". *IEEE Trans. Computers*, 24:940-941, 1975.
- [10] M. Sarfraz, A. Masood, M.R. Asim, "A New Approach to Corner Detection", *Proc. of International Conference on Computer Vision and Graphics*, 2004.
- [11] M. Sarfraz, M.F.A. Razzak, "A Web based system to capture outlines of Arabic fonts", *Information Sciences*, vol. 150(3-4), pp. 177-193, 2003.
- [12] M. Sarfraz, M.F.A. Razzak. "An Algorithm for Automatic Capturing of Font Outlines". *Journal of Computers & Graphics*, Elsevier Science, Vol. 26(5), 2002.