Surface Completion of an Irregular Boundary Curve Using a Concentric Mapping

William Martin and Elaine Cohen

Abstract. It is frequently necessary to complete the design of a surface from a specification of its boundary. This paper introduces a technique for completing the surface when the boundary is described by a non-self-intersecting, closed, planar, B-spline curve. The mapping produces a tensor product B-spline surface whose outer boundary is the input curve, and whose parameterization generalizes the polar parameterization of the disc.

§1. Introduction

In this paper we propose a new operator for generating a planar surface from a closed, non-self-intersecting piecewise polynomial boundary in the plane. We consider this approach to be a novel step towards the larger goal of surface completion from a free form curve boundary. This is a common problem arising in geometric modeling. Examples include "capping" extrusions and filling holes where adjacent patches come together. Holes also commonly occur in scanned datasets. There are many applications where such models must be made "watertight".

Given the importance of the problem, a number of methods have been proposed for surface completion. Rather than attempt to warp a rectangular *uv* domain to an irregularly shaped region, a common approach is to employ tensor product surfaces whose parameter domains are further restricted by trimming curves. Generally, the boundary must be densely sampled to accurately represent this subset and the parameterization originally associated with the boundary curve is lost. Moreover, the representation does not lend itself easily to further modeling operations.

Several methods have been introduced for hole-filling (e.g., [1]). Often these techniques do not address the parameterization of the filled region.

Curve and Surface Fitting: Saint-Malo 2002
XXX (eds.), pp. 1–10.

ISBN XXX.

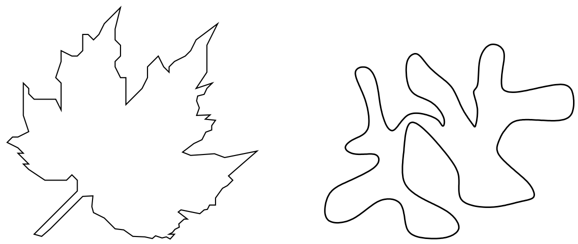

Fig. 1. Motivating examples for our work.

When the parameterization is addressed, it is often piecemeal, composed of a series of adjacent parametric patches.

Finally, there are a number of classic works on completing a surface from a series of bounding curves [2,4,6,8]. This work is most closely related to the algorithm we will develop. However, in contrast to our approach, these techniques generally assume the boundary can be naturally decomposed into n-faces, which can in turn be blended together. For example, schemes have been developed to complete a surface from 3, 4, 5, and 6-sided areas. There are many commonly occurring curve examples that do not easily admit such a decomposition (see Figure 1). Another drawback is that many techniques are tailored for a certain "sided-ness". Finally, the parameterization, when it is developed, is not always a straightforward mapping from a rectangular domain.

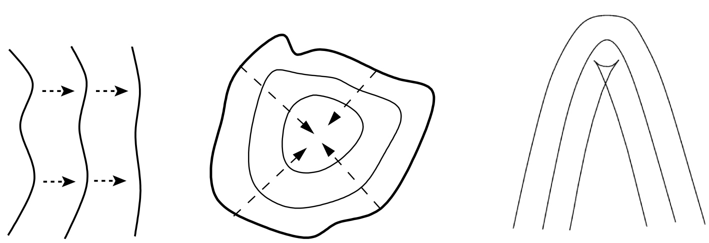

Fig. 2. Moving curves and a problem with offsets.

The starting point for surface completion algorithms is a description of the boundary. Therefore, an intuitive approach to completion is a parameterization that starts on the boundary and works its way inward (see Figure 2). This idea is evocative of offset curves. Take the sequence of curves generated by successively moving each point on the curve a fixed distance in the direction of its normal. The union of such a sequence can be used to parameterize the interior of the boundary. However, this type of completion has a problem in that all points do not generally come together simultaneously. Thus, as seen in Figure 2, portions of the offsets will begin to cross and must be clipped to avoid singularities in the parameterization. This results in a complex parameterization. Another possible technique is to use variable offsets. The problem now becomes how to choose the offset distances. This paper offers a solution — concentric parameterization.

A related technique that does not have the crossing problem is based on level set methods [7]. However, these methods are grid-based and do not produce parameterizations. Our technique uses parametric functions and therefore has straightforward application in most commonly used geometric modeling systems.

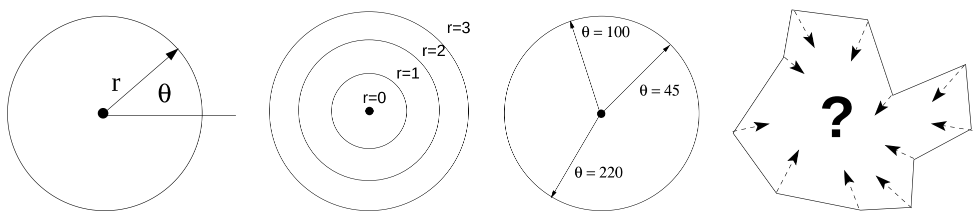

Fig. 3. The inspiration for the our parameterization.

Our parameterization is inspired by the standard (r, θ) parameterization of the disc. Such a surface parameterization respects the parameterization of the outer circle which is its boundary (see Figure 3). One parameter can be seen as traversing the boundary, whereas the other selects the successive scalings of the boundary which work their way to the center. The main goal of this paper is to find a method of surface completion which simulates this (r, θ)-type relationship while assuring that the successive offsets meet simultaneously (see Figure 3, rightmost).

The medial axis is the natural generalization of the circle's centerpoint to objects with more complex boundaries. The medial axis of a figure is defined to be the locus of the centers of all maximal inscribed circles. Such a circle will touch the boundary at at least 2 points. Since there is a corresponding point on the medial axis for each boundary point, it is natural to consider the trivial surface completion operator

$$(1-t)\gamma(s) + t\mathcal{MA}(\gamma(s))$$

where $\gamma(s)$ is a point on the boundary curve, and $\mathcal{MA}(\gamma(s))$ maps this point to the medial axis. The left frame of Figure 4 shows this mapping applied to a rectangle, and the middle panel shows the resulting surface parameterization.

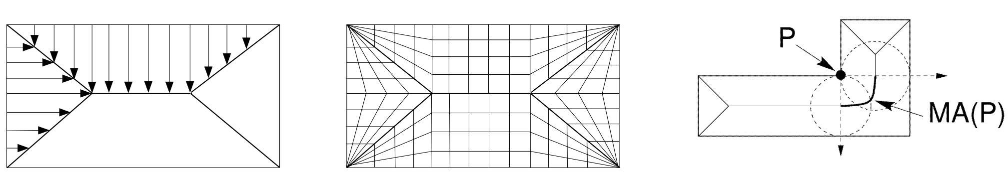

Fig. 4. Weaknesses in a direct medial mapping.

This parameterization is subject to a considerable distortion — in particular, every isoline (s, t_0) travels through the corners (Figure 4, middle). This certainly does not capture the intuitive notion of a disc-like parameterization. Another problem is that $\mathcal{MA}$ is not always a function. A concave vertex, for example, maps to an entire curve segment along the medial axis (Figure 4, right).

§3. The Concentric Parameterization

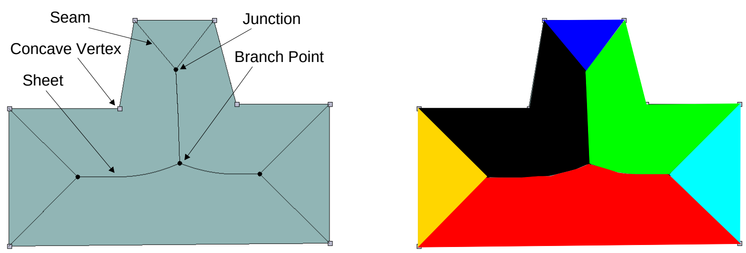

Fig. 5. Nomenclature review; regions formed by the medial axis.

Figure 5 contains a brief review of nomenclature. The medial axis can be divided into roughly two types of curves. *Sheets* are portions of the medial axis that do not touch the boundary. They are joined to the boundary (and in particular to the convex points) by the *seams*. It was the inclusion of these seams in the surface completion that led to the severe distortions of Figure 4. Our new mapping projects solely to the sheet, using the seams to guide the contraction.

§3.1 Polygonal boundary

Let us first consider the case of a polygonal boundary. The right panel of Figure 5 illustrates that the medial axis divides the polygon into regions. Each region is bounded by two seams, a portion of the boundary curve, and a portion of the sheet (which can degenerate to a point). Each point contained in a region is closer to its boundary and sheet segment than it is to any of the other boundary or sheet segments. In particular, the boundary segment is closer to the part of the sheet bounding its region than it is to any other part of the sheet. (Similar observations have been made by [3]). Thus it is natural to consider a mapping of each region boundary onto its corresponding sheet segment. Our contraction is based on a particular approximation to the sheet:

Definition 1. *The* concentric axis *of a polygon is a piecewise linear approximation to the medial axis sheet whose vertices, termed* concentric vertices *satisfy the following two properties:*

a) *every vertex of the polygon corresponds to a concentric vertex and*
b) *every concentric vertex has a corresponding vertex on the boundary polygon of each region it borders.*

If condition b) is not satisfied, the surface completion may contain holes [3]. The set of concentric vertices will include the junction and branch points (Figure 5). The convex vertices naturally map to the junction points of the medial axis. To fully satisfy condition a), we must find a mapping of the concave boundary vertices onto the medial axis. Considering the right pane of Figure 4, it is reasonable to select any point in the range of $\mathcal{MA}$ corresponding the concave vertex as an addition to the concentric axis. If there are preexisting concentric vertices in this range, we may select the closest one to simplify the next step (see Figure 8(4-5)).

We now form a curve from the concentric axis; this curve will be blended with the boundary curve to complete the surface.

Definition 2. *The* concentric control polygon *or more simply* concentric polygon *is a sequence of concentric vertices determined by traversing the boundary polygon in the direction of increasing parametric value, and inserting the concentric vertex corresponding to each boundary point encountered.*

Definition 3. *A* concentric curve *is a piecewise linear B-spline defined by a concentric control polygon and a corresponding knot vector (termed the* concentric knot vector*).*

We want to preserve the original parameterization of the boundary curve as an isoparametric direction in the surface completion. Since even the case of a polygonal boundary admits a non-uniform parameterization, we assume that there is a knot vector associated with the boundary. Each vertex of the boundary therefore has an associated parameter, which is used to assign parameters to the corresponding concentric vertices, and form a knot vector for the concentric curve.

In order to satisfy condition b) of Definition 1, it is necessary to ensure that each concentric vertex has a boundary vertex correspondence for each region it borders. If a boundary correspondence is lacking, one will be inserted as follows. Because we have imbued the concentric curve with a parameterization we can approximate the parameter value of the concentric vertex for this region, and add this value to the concentric knot vector. We add the concentric vertex to the corresponding location in the concentric polygon. When we bring the boundary curve and concentric curve into the same spline space for the final blend, the refinement will find the boundary correspondence automatically. The mapping based on this blend is demonstrated for a rectangle in Figure 6. Other examples are shown in the left panes of Figures 9, 10, 12, and 13.

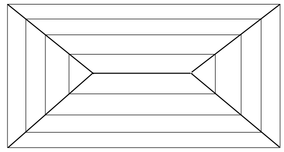

Fig. 6. Parameterization of the rectangle which results from our mapping.

§3.2 Generalization to higher order curves

Direct application of this technique to arbitrary curves is somewhat difficult. One wishes to perform the sort of contraction introduced above on a finite number of points, but one also generally wants to avoid discretizing the curve. The B-spline representation provides a tractable solution to this problem. Because the B-splines are a) defined by a control polygon, b) possess the convex hull property with respect to the control polygon and c) are variation diminishing with respect to the control polygon, the contraction of the boundary control polygon onto its concentric polygon implies a mapping of the boundary curve to the concentric curve. Since the boundary control polygon converges to the boundary curve under refinement, the medial axis of the control polygon will converge to that of the boundary curve in the limit. Hence, the technique of the previous section provides a good approximation to the continuous case with sufficient refinement. The quality of the parameterization is largely dependent on how well the boundary control polygon approximates the boundary curve.

1) Assign a parameterization to the boundary vertices using the nodal values (Greville abscissas) of the spline space (Fig. 8(2)).
2) Calculate the medial axis of the control polygon (Fig. 8(3)).
3) For each concave vertex, insert a concentric vertex (Fig. 8(4-5)).
4) Form the concentric axis control polygon: for each boundary vertex, find the closest concentric vertex along the seam, and add it to the concentric axis. Add the associated parameter (nodal value from step 1) to the concentric knot vector (Fig. 8(6-12)).
5) For each region, if there are internal concentric vertices, insert them at the appropriate place in the concentric control polygon. Calculate the interpolated parameter value, and add it to the concentric knot vector (Fig. 8(13-14)).
6) Degree raise the concentric linear B-spline to match the degree of the boundary.
7) Refine the boundary and concentric curves using the union of their knot values.
8) Form the sweep surface (Fig. 8(15)).

Fig. 7. Basic concentric completion algorithm.

The concentric completion algorithm for curves is an extension of the algorithm for polygons. The method of determining the parameterization of the concentric curve is a straightforward generalization. We associate with each boundary vertex the nodal value of its associated B-spline basis function. This is the parameter where, to first approximation, its influence is greatest. We summarize the concentric algorithm in Figure 7. Figure

8 demonstrates the algorithm on a simple, uniform, cubic curve (Figure 8(1)).

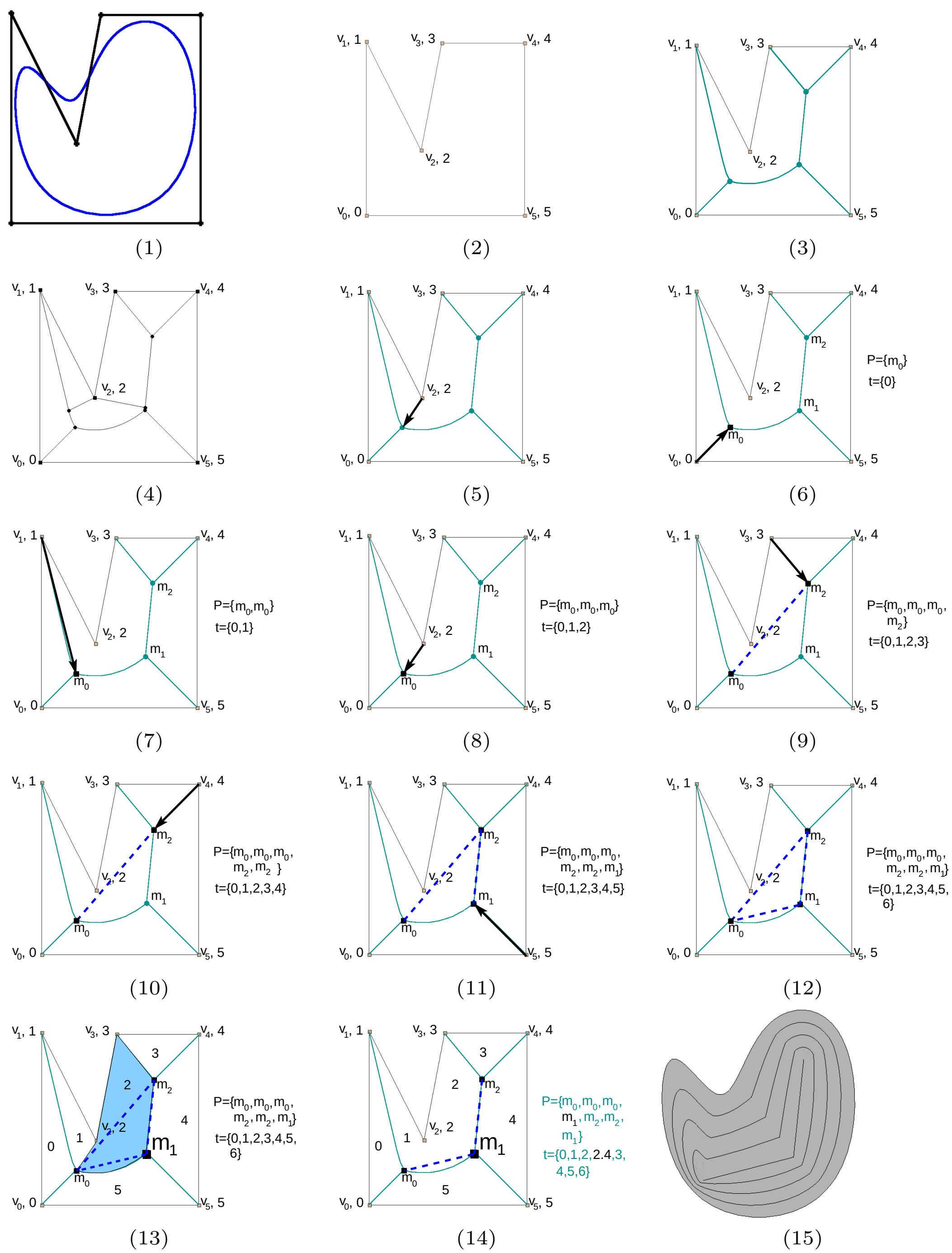

Fig. 8. The concentric completion algorithm, illustrated.

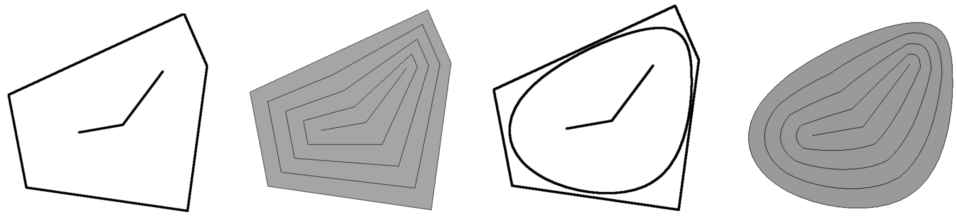

Fig. 9. Our algorithm applied to a simple convex curve (degrees 1 and 3).

Figure 10 shows a simple case where using the concentric axis of the boundary control polygon fails, because this approximate skeleton crosses the boundary curve. This example violates our assumption that the boundary control polygon is a good approximation to the shape of the underlying curve. However, we can detect cases where the concentric curve crosses the boundary curve rather easily. We simply test each concentric polygon edge for crossings. One way to do this is to cast a ray along each edge. If there is an intersection with the boundary curve between the endpoints of the segment, then the concentric axis is not valid. We note that for low order curves, this intersection can be accomplished analytically. A similar method can be used to determine the quality of the approximation by calculating distance to the boundary curve [5].

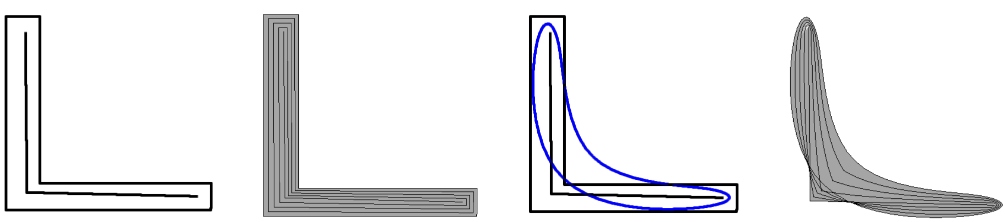

Fig. 10. An example where the basic algorithm fails (degrees 1 and 3).

If the concentric axis approximation is found inadequate, one option is to refine the boundary control polygon, and restart the concentric algorithm. However we want to avoid an unnecessary explosion in the number of surface control points. An alternative is to move the concentric curve until it is seated within the boundary curve. It is sometimes possible to do this by calculating the medial axis of the refined control polygon and moving the concentric axis into alignment. When successful, this can eliminate the need for additional control points. This technique was used in Figure 11 and the right frames of Figure 13. These figures also demonstrate the tradeoff in the approximation quality and the number of control points.

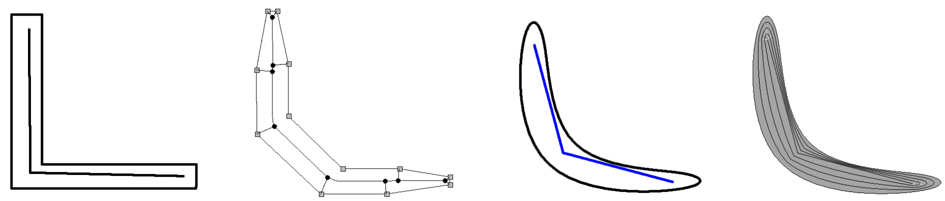

Fig. 11. Moving the coarse concentric axis to its refined position.

§4. Conclusions

We have presented a technique for completing a surface from a non-self-intersecting, closed, planar B-spline curve. Figures 12 and 13 demonstrate the algorithm on some more complicated curves. Our method has produced reasonable parameterizations of a variety of complex figures where existing completion techniques would fail or experience difficulty. Presently we pursue a generalization of the technique to non-planar boundary curves and methods for accommodating further modeling operations involving the boundary and the completed surface. The present technique is not ideal for boundaries with detail at many scales. Such boundaries tend to have secondary and tertiary branches that have little to do with the basic shape of the surface. We are developing methods to deal with these issues.

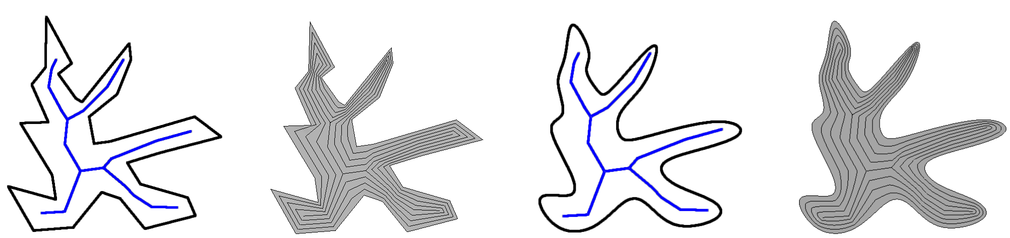

Fig. 12. A more complicated example for degrees 1 and 3.

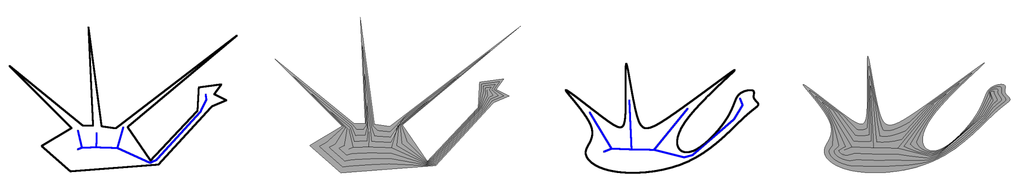

Fig. 13. A sharp polygon for degrees 1 and 3.

Acknowledgments. The authors thank Amy and Bruce Gooch, Richard F. Riesenfeld, Peter Shirley, and Michael M. Stark for their thoughtful feedback and encouragement. This research was supported by NSF grant IIS 0218809 and the STC for Graphics and Visualization (EIA-89-20219).

References

1. Chui, Charles K. and Min-Jun Lai, Filling polygonal holes using C^1 cubic triangular spline patches, Comput. Aided Geom. Design **17** (2000) 297–307.
2. Coons, S. A., Surface patches and B-spline curves, in *Computer Aided Geometric Design* (ed. Barnhill and Riesenfeld), 1–16, Academic Press, Orlando, FL, 1974.
3. Kimmel, Ron, Doron Shaked, Nahum Kiryati, and Alfred M. Bruckstein, Skeletonization via Distance Maps and Level Sets, in Computer Vision and Image Understanding **62**(3) 382–391.
4. Loop, Charles and Tony DeRose, Generalized B-spline surfaces of arbitrary topology, in Computer Graphics (Proceedings of SIGGRAPH 90) **24**(4) 347–356.
5. Lutterkort, David and Jörg Peters, Tight linear envelopes for splines, Numer. Math. **89** (2001) 735–748.
6. Sabin, M. A., Non-rectangular surfaces suitable for inclusion in a B-spline surface, in Proceedings of Eurographics '83, 57–69.
7. Sethian, J. A., *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, New York, 1999.
8. Zheng, J. J. and A. A. Ball, Control point surfaces over non-four-sided areas, Comput. Aided Geom. Design **14** (1997) 807–821.

William Martin
School of Computing
University of Utah
Salt Lake City, UT 84112
`wmartin@cs.utah.edu`
http://www.cs.utah.edu/∼wmartin

Elaine Cohen
School of Computing
University of Utah
Salt Lake City, UT 84112
`cohen@cs.utah.edu`
http://www.cs.utah.edu/∼cohen