A Philosophy for Smooth Contour Reconstruction

John K. Johnstone and Kenneth R. Sloan*

Abstract

We propose an approach to the reconstruction of point samples into smooth models based upon the generation of isocurves, with concentration on contour data. We show how to build an isocurve through any point and how to choose which isocurves are computed. The isocurves are used to build a tensor product surface for each component of the model isomorphic to a cylinder.

Keywords: surface reconstruction, contour, isocurves.

1 Introduction

The conversion of point samples from a known object into a smooth, three-dimensional, geometric model of the object is receiving much current attention, particularly for its applications in medicine and model creation for animation. An important example of this reverse engineering problem is the conversion of ordered, planar contour data, segmented from biomedical CT/MR images of patient anatomy, into smooth geometric anatomical models (Fig. 4a,9). In this paper, we propose an approach to the reconstruction of point samples into smooth models based upon the generation of isocurves (isoparametric curves). Although this approach can be applied to general reconstruction problems, we discuss it here in the context of the smooth reconstruction of contour data.

The smooth reconstruction of point samples can be viewed as a two step process: the identification of regions of the underlying object isomorphic to a cylinder, which we call **metatubes** (Figure 5), followed by the modeling of these regions through the construction of isocurves (Figure 6). Since it is important to preserve smoothness between the regions, the modeling of one region is not entirely independent of the modeling of another region. However, a discussion of the issues arising from the interdependency of regions is beyond the scope of this paper (see [5]) and we concentrate here on the independent modeling of a single region. Isocurves suffice for this modeling because of tensor product interpolants. A tensor product interpolant [1] takes as input a rectangular grid of points, and it both interpolates the points *and* imposes isocurves according to the grid, since a row (or column) of the input grid becomes an isocurve of the surface. Thus, if we determine a set of valid isocurves (both vertical and horizontal) for an object we can create a surface that respects these isocurves using the tensor product interpolant. Actually, since the input to a tensor product interpolant is simply a grid of points, the grid of isocurves can be replaced by an isogrid of points.

Definition 1 *An* **isogrid** *is a rectangular grid of points such that all points of the same row (or column) lie on the same isoparametric curve.*

*Department of Computer and Information Sciences, The University of Alabama at Birmingham, 115A Campbell Hall, 1300 University Boulevard, Birmingham, AL 35294 USA. {johnstone,sloan}@cis.uab.edu

In short, we can reduce the reconstruction problem to the identification of metatubes followed by the construction of an isogrid over each metatube. A useful interpretation of the isocurves is that they impose an order on the unorganized point data, in preparation for a tensor product interpolant.

A tensor product interpolant is the appropriate smooth model for a metatube because the rectangular structure of the tensor product surface is naturally suited both to the cylindrical topology of the metatube and the particularly rectangular structure of contour data over a metatube, with its rows of data. Just as a triangular spline surface reflects the topology of a triangle and may be most appropriate for methods that build patches through the smoothing of individual triangles of a triangulation [7], the tensor product surface reflects the topology of the metatube and the global nature of our modeling process, where isocurves are constructed across the entire metatube. Serendipitously, the tensor product surface is also widely supported in modeling systems, so the methods described in this paper can be easily incorporated into existing software.

The identification of metatubes, regions of the object isomorphic to a cylinder, is basically the identification of the object's topology and has received considerable discussion in the literature. For contour data, the identification of metatubes is equivalent to the identification of branching, and there are many approaches such as the use of minimum spanning trees, the use of contour overlap, and cylinder growing [8]. We will not further explore this issue here. The main subject of this paper is the second problem: the construction of a model over a metatube using isocurves. The rest of the paper is structured as follows. Section 2 shows how to build an isocurve through any point, using a triangulation of the data as a guide. It includes a motivation of why this method yields a good approximation to the 'true' isocurve. Sections 3 and 4 are more philosophical in nature. Section 3 discusses the necessity for contour resampling and Section 4 argues that it is important to be able to model the data to a controllable tolerance. Finally, Section 5 shows how to choose which isocurves are computed, guided by the desired tolerance.

2 Isocurves from a triangulation

Given a point $\mathbf{P}$ on a metatube, we wish to determine another point on $\mathbf{P}$'s isocurve. Since we do not yet have a surface model for $\mathbf{P}$'s metatube, what do we even mean by $\mathbf{P}$'s isocurve? We will use a triangulation of the metatube's contour data (Figure 5), which is an efficient approximation to the surface model of the metatube. Indeed, previous work on the reconstruction of contour data has assumed that a triangulation *is* the surface model. We are not content to stop with a triangulation as the final model because of the improved accuracy of smooth models. (By accuracy, we mean fidelity to the true shape of the underlying object. [4] contains some evidence of the improved accuracy of smooth models, and we will document the improvement further in a forthcoming paper.) However, we are happy to use a triangulation as an aid to the construction of isocurves.

There are many different triangulation methods, and the accuracy of the isocurve will naturally depend on the accuracy of the triangulation. We leave the choice of triangulation as a free parameter, since interesting triangulation techniques continue to be created and there is probably no one method that is always superior. In our work, we use the classical minimal area triangulation of Fuchs, Kedem and Uselton [2], an excellent choice for typical contour data. We will now establish that a triangulation is a good source of information about isocurves, restricting our attention to contour data.

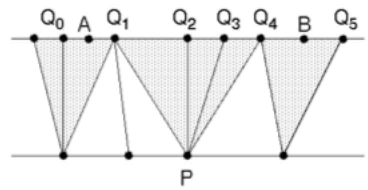

Figure 1: Fan at P

A triangulation is an approximation to a certain smooth surface. In particular, as the number of data points increases (uniformly across each contour) and the spacing between contours decreases (again uniformly), the triangulation converges to a smooth surface. This smooth surface, the limit of the triangulations, is a desirable model of the metatube as long as the triangulation method itself yields accurate discrete models (up to the level of the available data). For example, optimal properties (such as minimal surface area) will be inherited from the triangulation to the smooth limit surface.

Of course, we do not have access to the limit surface, only the triangulation. However, the triangulation can be used to get a good approximation to the isocurves of the limit surface. The convergence of a triangulation to a smooth surface can be split into two steps: as the number of data points increases (but the spacing between contours remains constant), the triangulation between two contours converges to a lofting [1] between the contours; then as the spacing between contours decreases, the collection of loftings (between each adjacent pair of contours) converges to a smooth surface. Let us first consider the triangulation as a discrete approximation of lofting.

In the triangulation, connecting a point $\mathbf{P}$ with the points $\mathbf{Q_1}, \ldots, \mathbf{Q_k}$ ($k >= 2$) of a neighbouring contour is tantamount to saying that, in the lofting, $\mathbf{P}$'s corresponding point on the neighbouring contour lies between $\mathbf{Q_1}$ and $\mathbf{Q_k}$.[1] See Fig. 1 where $k = 4$. In the absence of other information, the most reasonable guess at the exact position of the point corresponding to $\mathbf{P}$ in the lofting is the midpoint of $\overset{\frown}{\mathbf{Q_1 Q_k}}$.

Remark 2.1 *We assume that the data points of a contour have been interpolated by a spline curve so that we have a contour curve rather than discrete contour points (see Section 3). Therefore, it is meaningful to talk about the arc* $\overset{\frown}{\mathbf{Q_i Q_j}}$ *between two data points* $\mathbf{Q_i}$, $\mathbf{Q_j}$.

Since the lines of the lofting converge to the isocurves of the smooth limit surface as the distance between contours decreases, the midpoint of $\overset{\frown}{\mathbf{Q_1 Q_k}}$ is also a good guess at a point on the same isocurve as $\mathbf{P}$. We now formally define this estimate, called the isopartner.

[1] Actually, the precise range is slightly more relaxed: the corresponding point must lie between $\mathbf{A}$ and $\mathbf{B}$, where $\mathbf{A}$ is the midpoint of $\overset{\frown}{\mathbf{Q_0 Q_1}}$ and $\mathbf{B}$ is the midpoint of $\overset{\frown}{\mathbf{Q_k Q_{k+1}}}$. Certainly, if $\mathbf{P}$'s corresponding point lies outside this interval, $\mathbf{P}$ would be connected to other points on the neighbouring contour in the triangulation (such as $\mathbf{Q_0}$ or $\mathbf{Q_{k+1}}$). However, using the slightly larger interval $\overset{\frown}{\mathbf{AB}}$ rather than $\overset{\frown}{\mathbf{Q_1 Q_k}}$ leads to overlapping ranges, which can lead to problems such as intersecting isocurves.

Definition 2 *Let* $\mathbf{C_1}$ *and* $\mathbf{C_2}$ *be adjacent contours in a metatube, represented by contour curves* $\mathbf{C_1}(s)$ *and* $\mathbf{C_2}(t)$. *Let* $\mathbf{P} \in \mathbf{C_1}(s)$ *be a vertex of the triangulation that is connected with* $\mathbf{Q_1} = \mathbf{C_2}(t_1), \ldots, \mathbf{Q_k} = \mathbf{C_2}(t_k)$ *on* $\mathbf{C_2}$ *(*$k >= 2$*). The* **isopartner** *of* $\mathbf{P}$ *on* $\mathbf{C_2}$ *is* $\mathbf{C_2}(\frac{t_1+t_k}{2})$. *The edges* $\mathbf{PQ_1}$ *to* $\mathbf{PQ_k}$ *form a* **fan** *at* $\mathbf{P}$. $\mathbf{P}$ *is the* **base** *of this fan.*

Since we use a centripetal parameterization of the contour curve, the isopartner of $\mathbf{P}$ is close to the geometric midpoint of $\widehat{\mathbf{Q_1 Q_k}}$. We now define the isopartner for the remaining points of the contour curve, in terms of the isopartners of Definition 2.

Definition 3 *Let* $\mathbf{C_1}$ *and* $\mathbf{C_2}$ *be adjacent contours in a metatube, represented by contour curves* $\mathbf{C_1}(s)$ *and* $\mathbf{C_2}(t)$. *Let* $\mathbf{P} = \mathbf{C_1}(\bar{\mathbf{s}})$, $\mathbf{P}$ *not the base of a fan of the triangulation to* $\mathbf{C_2}$. *Let* $\mathbf{C_1}(s_1)$ *and* $\mathbf{C_1}(s_2)$ *be the bases of the two fans that surround* $\mathbf{P}$, *with isopartners* $\mathbf{C_2}(t_1)$ *and* $\mathbf{C_2}(t_2)$. *The* **isopartner** *of* $\mathbf{P}$ *on* $\mathbf{C_2}$ *is* $\mathbf{C_2}(\bar{\mathbf{t}})$ *where* $\bar{\mathbf{t}} = (\frac{s_2-\bar{\mathbf{s}}}{s_2-s_1} * t_1 + \frac{\bar{\mathbf{s}}-s_1}{s_2-s_1} * t_2)$. *That is,* $s_1 : \bar{\mathbf{s}} : s_2 = t_1 : \bar{\mathbf{t}} : t_2$.

To build a set of points on the isocurve through $\mathbf{P}$, we follow isopartners up and down from $\mathbf{P}$, through all contours of the metatube (Figure 7). A column of the metatube's isogrid (Definition 1) will consist of the points $\mathbf{P}$, $\mathbf{P}' = \text{isopartner}(\mathbf{P})$, $\mathbf{P}'' = \text{isopartner}(\mathbf{P}')$ and so on. Notice that $\mathbf{P}$ is an *arbitrary* point on the contour curve of a contour in the metatube. Isopartners should yield a good isocurve because a fan in the triangulation is a discrete approximation to a line in the limit lofting, which is a linear approximation to an isocurve in the limit surface, which is a desirable model for the metatube.

Definition 4 *Each isocurve is defined by a set of points, called* **beads**, *one per contour. The* **seed** *of an isocurve is the first bead that was chosen, from which the other beads are created by following isopartners.*

We have described how to build vertical isocurves. Horizontal isocurves are also needed, since there is no reason that the contours, which are arbitrarily collected from the object, should be the horizontal isocurves. We also want to control the density of the horizontal isocurves (Section 5). The computation of horizontal isocurves is basically the same procedure as the computation of vertical isocurves. The 'contours' are the vertical isocurves, which are sampled into data points and then triangulated. Isocurves are computed by following isopartners across this triangulation, the only difference being that care must be taken to guarantee that the resulting isocurves are closed (see [3]).

3 Resampling

The quality of the isocurves produced by a triangulation decreases as the point sampling of the contours decreases, since the triangulation's approximation to a lofting degenerates. We are particularly eager to avoid unusually large gaps between samples, since any wide fan has a high probability of yielding an inaccurate isopartner (Figure 2 and Example 3.1). Large gaps are quite common in contour data, since flat regions (regions of low curvature) will usually be sparsely sampled.

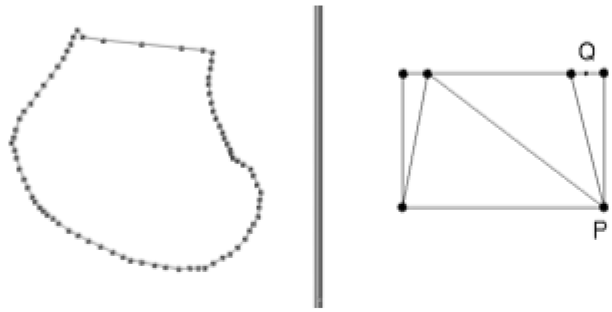

Figure 2: (a) Contour with sparse sampling in flat region (b) Possible resulting error

Example 3.1 *Figure 2a shows an example of a contour, sampled manually from a CT image of the distal femur, that illustrates sparse sampling in flat regions. Figure 2b shows an abstraction of this type of data. The computed isopartner of* **P** *will be pushed far to the left by the sparse sampling over the flat region, even if the true point corresponding to* **P** *in the lofting should be* **Q**.

We emphasize that even when sparse samples yield a good definition of the contour shape, they will lead to bad isocurve selection.

Fortunately, this type of problem from coarse point sampling can be solved by resampling, as we will now explain. (The other source of error in isocurves, a large distance between contours, is inherent to the reconstruction problem and is much harder to correct.)

Historically, the entire contour is represented as the piecewise linear curve connecting the data points. However, this is largely an artifact of the reconstruction of the contours by a triangulation. Since the true shape of most contours (and certainly all biological contours) is smooth, and due to the superiority for the modeling of shape of higher degree (say cubic) interpolating curves over piecewise linear interpolating curves, we propose the following axiom:

Axiom 1: The best model of a contour is a smooth (interpolating) curve.

If image segmentation was done properly, this contour curve will be a close approximation to the true shape of the object in this slice. The data points on the contour curve can be viewed as a particular choice of sampling of the contour curve. Moreover, there is nothing particularly special about this sampling: although sampling is sometimes related to curvature (fewer samples in a region of low curvature), it is largely arbitrary, especially when the point data was obtained from manual image segmentation. This leads to another axiom:

Axiom 2: Data points can be resampled from the contour curve of Axiom 1.

We embrace a policy of data refinement, such as contour resampling, rather than burdening the algorithm with the idiosyncracies of the original data throughout the entire reconstruction process.

Figure 3: Contours from endocardial and epicardial walls of heart's left ventricle

The only restriction is that the refinement be consistent and well-founded. In particular, this resampling strategy is applied to fill in any large gaps in the contour data, thus removing errors such as described above.

Remark 3.1 *If there is a concern that original data points may represent features of the object that should be preserved, we can resample so that all original data points are unchanged but other data points are added.*

4 Approximation within a tolerance

If we wish to exactly interpolate the data, an isocurve must be built through each data point. However, an approximating surface is more appropriate for most contour data. This is motivated by the imperfection of contour data. Consider contours from medical imaging. These contours are derived from segmentation of a CT or MR image. Segmentation methods vary from completely manual to semi-automated [6, 9] to completely automated, however all are error-prone. Automated methods are regularly fooled by the image, and manual methods rely on manual dexterity, domain knowledge, time and patience. For example, the contours in Figure 3 were captured semi-automatically by a knowledgeable professional who regularly contours MR images for an MR lab, yet considerable surface detail (papillary muscle) has been captured that may be irrelevant for analysis, and there are small errors (e.g., endocardial contours occasionally pierce the epicardial contours). Therefore, an approximating surface is appropriate.

Our method has control over the level of approximation of the point data. The model will approximate the point data within a user-specified tolerance: i.e., the distance from each data point to the model will be less than the tolerance. This is accomplished through the choice of isocurves (see Section 5). The tolerance can be chosen, for example, based on the accuracy of the segmentation method. Tolerance-based approximation blurs the distinction between approximating and interpolating models. It offers a useful mechanism for hierarchical modeling of the object, using progressively smaller and larger tolerances (Fig. 8-9). Surface detail can be removed or added at the user's discretion.

5 Which isocurves?

The remaining question is how many isocurves should be constructed, and where? The user-defined tolerance drives the choice of isocurves on a metatube. The isocurves must define a patch that reconstructs the data points within the tolerance. Since we prefer concise surface models, we also want to minimize the number of isocurves.

Reconstruction within the tolerance ϵ can be expressed by the following condition:

For all contours C_i with data points $P_{i1}, \ldots, P_{in_i}$, $\text{dist}(P_{ij}, \text{bead curve of } C_i) < \epsilon$.

where the bead curve of contour C_i represents the current reconstruction of C_i in the smooth surface.

Definition 5 *The* **bead curve** *of a contour C is the curve that interpolates the present set of beads on C (one bead from each isocurve).*

Suppose that we have seeded a certain number of isocurves, and want to know whether to stop. We have the following algorithm for building isocurves.

- Compute $\text{dist}(P_{ij}, \text{bead curve of } C_i)$, for all P_{ij} and all contours C_i.
- If one or more distances are greater than the tolerance ϵ, choose the largest distance and use this data point to seed an isocurve (Definition 4).
- Repeat until no distances are greater than ϵ.

At added expense, one can replace the maximal distance from the data points to a bead curve by the maximal distance from the entire contour curve to the bead curve. To minimize the number of isocurves, we also favour the seeding of data points on short curve segments (i.e., curve segments whose perimeter is small relative to the contour perimeter) since they are less likely to be covered by other isocurves.

6 Conclusions

The construction of a smooth model from contours can be reduced to the problem of the proper choice of isocurves. A small set of accurate isocurves can be found, after some resampling of the contour data, from a triangulation of the data. The resulting model respects the point data up to a user-defined tolerance (including zero tolerance).

References

[1] Farin, G. (1997) Curves and Surfaces for Computer Aided Geometric Design. Academic Press (Boston), 4th edition.

[2] Fuchs, H., Z. Kedem and S. Uselton (1977) Optimal Surface Reconstruction from Planar Contours. CACM 20(10), 693–702 (also SIGGRAPH 77, San Jose).

[3] Johnstone, J.K. and K.R. Sloan (1995) Tensor Product Surfaces Guided by Minimal Surface Area Triangulations. Visualization ’95, Atlanta, 254–261.

[4] Johnstone, J.K. and K.R. Sloan (1995) Visualization of LV Wall Motion and Thickness from MRI Data. Computers in Cardiology ’95, Vienna, 249–252.

[5] Johnstone, J.K. and K.R. Sloan (1997) Smooth Reconstruction of Branching Contours, under preparation.

[6] Kass, M., A. Witkin and D. Terzopoulos (1988) Snakes: Active Contour Models. International Journal of Computer Vision 1(4), 321–331.

[7] Mann, S., C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, and K. Sloan (1992) A Survey of Parametric Scattered Data Fitting using Triangular Interpolants. In *Curve and Surface Design*, H. Hagen, ed., SIAM, Philadelphia, 145–172.

[8] Meyers, D., S. Skinner and K. Sloan (1992) Surfaces from Contours. ACM Transactions on Graphics 11(3), July 1992, 228–258.

[9] Mortensen, E. and W. Barrett (1995) Intelligent Scissors for Image Composition. SIGGRAPH ’95, Los Angeles, 191–198.

[10] Shinagawa, Y. and T. Kunii (1991) The Homotopy Model: A Generalized Model for Smooth Surface Generation from Cross Sectional Data. Visual Computer 7, 72–86.

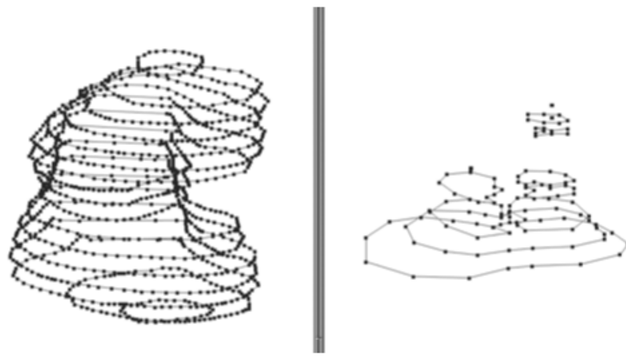

Figure 4: Contour data: (a) medical (b) topographic

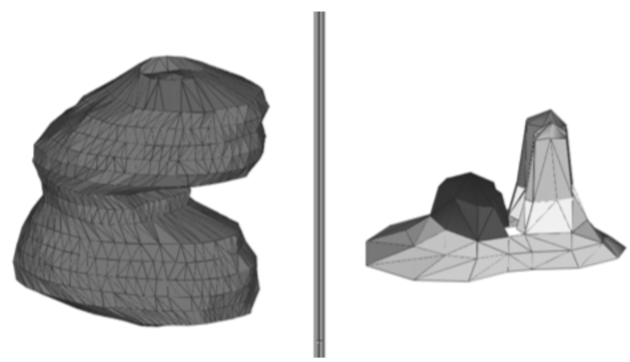

Figure 5: Triangulation (coded by metatube)

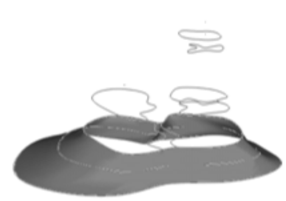

Figure 6: Smooth model of a single metatube

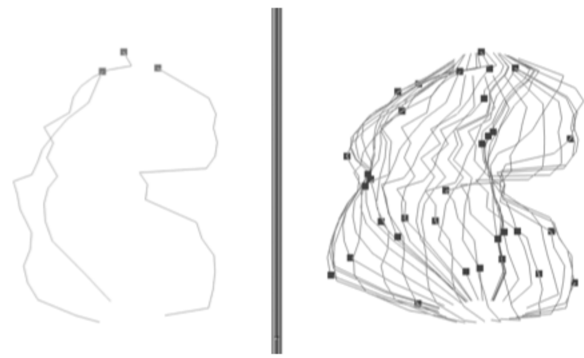

Figure 7: Vertical isocurves, with seeds: (a) large tolerance (b) small tolerance

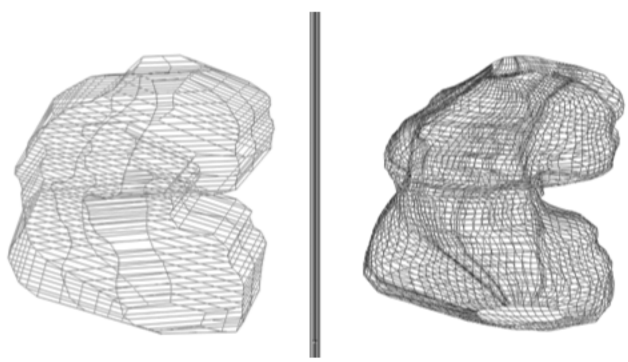

Figure 8: Tensor product control meshes: (a) large tolerance (b) small tolerance

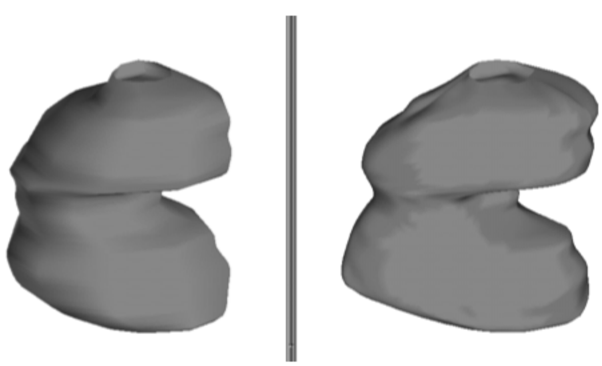

Figure 9: Smooth model of distal femur: (a) large tolerance (b) small tolerance