ELECTRONIC SKELETONS:
MODELING SKELETAL STRUCTURES
WITH PIECEWISE ALGEBRAIC SURFACES

Chanderjit L. Bajaj

CSD-TR-91-076
August 1991

Electronic Skeletons: Modeling Skeletal Structures with Piecewise Algebraic Surfaces

Chanderjit L. Bajaj *
Department of Computer Science,
Purdue University,
West Lafayettte, IN 47907

Abstract

We present two algorithms to construct C^1-smooth models of skeletal structures from CT/NMR voxel data. The boundary of the reconstructed models consist of a C^1-continuous mesh of triangular algebraic surface patches. One algorithm first constructs C^1-continuous piecewise conic contours on each of the CT/NMR data slices and then uses piecewise triangular algebraic surface patches. to C^1 interpolate the contours on adjacent slices. The other algorithm works directly in voxel space and replaces an initial C^0 triangular facet approximation of the model with a highly compressed C^1-continuous mesh of triangular algebraic surface patches. Both schemes are adaptive, yielding a higher density of patches in regions of higher curvature.

*Supported in part by NSF grant CCR 90-02228 and AFOSR contract 91-0276

1 Introduction

Skeletal model reconstruction from voxel data has been an active research area for many years. Schumaker [15] presents an excellent survey of the wide range of techniques that have been utilized. Curiously, and as he points out, the majority of the reconstruction techniques produce only planar C^0 approximations of the data set. Relatively little has been achieved in constructing smooth skeletal models using curved surface patches.

There are primarily two classes of model reconstruction techniques. One class of methods first constructs planar contours in each CT/NMR data slice and then connects these contours by a triangulation in three dimensional space. The triangulation process is complicated by the occurrence of multiple contours on a data slice (i.e. branching). Early contributions here are by Keppel [8], Fuchs, Kedem and Uselton [6]. The optimal algorithms due to Keppel, Fuchs et. al work by computing a graph in which each node represents a spanning arc and each edge in the graph represents a triangle defined uniquely by two spanning arcs that share a point. A shortest path algorithm is used to find the path that corresponds to the triangulation of minimum weight. Instead of planar contours one may compute a C^1 continuous piecewise curve approximation in each of the data slices. Examples of such techniques are [14] using conic splines and [13] using parametric B-splines. The stack of contours are then interpolated or least square approximated using piecewise tensor splines [10, 11] or nontensor piecewise smooth surfaces [5].

The other class of methods uses a hierarchical subdivision of the voxel space to localize the triangular approximation to small cubes [9]. This method takes care of branching, however the local planar approximation based on the density values at the corner of the subcube may sometimes be ambiguous. An extension of this scheme which computes a C^1 piecewise quadratic approximation to the data within subcubes is given in [12]

In this paper we present an algorithm belonging to each of the above classes, to construct C^1-smooth models of skeletal structures from CT/NMR voxel data. The boundary of the reconstructed models consist of a C^1-continuous mesh of triangular algebraic surface patches. For definitions and basic manipulations of algebraic surfaces, the reader is referred to [2]. Section 2 describes the first algorithm which starts by constructing C^1-continuous piecewise conic contours on each of the CT/NMR data slices and then uses piecewise triangular algebraic surface patches, of degree at most seven, to C^1 interpolate the contours on adjacent slices. Section 3 describes the other algorithm which works directly in voxel space and replaces an initial C^0 triangular facet approximation of the model with a highly compressed C^1-continuous mesh of triangular algebraic surface patches, of degree at most seven. Both schemes are adaptive, yielding a higher density of patches in regions of higher curvature.

2 Working with 2D Slices

Our first step is to compute a polygonal contour on each data slice. The raw voxel scan data is in the form of a two dimensional array of two byte integers, one array for each planar slice through the object. The value in each cell of the array is related to the density of the scanned object at that point in space. Each array may contain any number of cross sections, i.e., each slice may cut the scanned object in multiple places. To locate the cross sections within a two dimensional slice the following simple algorithm is employed: (1) scan for a cell on an initial edge, (2) starting at this cell hug the exterior of the cross section working from cell to cell and creating a list of two dimensional points until

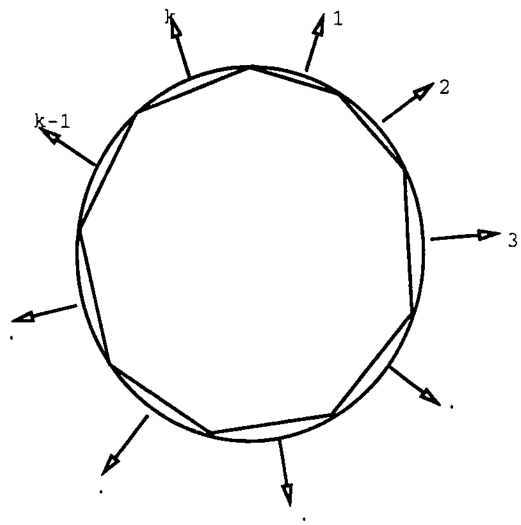

Figure 1: A regular subdivision of the space of normals on a planar contour

the beginning is reached or a dead end is found, (3) if a dead end is found backtrack, (4) if the path closes and the algorithm does not backtrack to the beginning point then smooth and compress the list of points if necessary. In our implementation of this algorithm the following heuristic rule was used: if the density value in a cell c is within range and if the density values of all the cells surrounding c are within range, then the cell c is *acceptable.* The point list is smoothed and compressed by growing segments that are within a predescribed constant value of the original polyline.

The next step is to compute a C^1-continuous piecewise conic approximation of the polygonal contour on selected data slices. The data slices are partitioned into groups such that the geometry of the contours within a group is fairly homogeneous. Smaller sized groups are chosen for instance, around skeletal regions of medical importance i.e. close to ligament attachments and where primary nerves pass close to the bone. A single data slice is selected from each group. Any of the algorithms discussed in [14] can be used to compute a C^1-continuous piecewise conic approximation of the polygonal contour on the selected data slices An alternate and perhaps simpler scheme for the placement of the conic segments is as follows. The points on a unit circle are in one-to-one correspondence with the normal directions, (or alternatively the slopes) of the line segments which make up the polygonal contour. Consider any regular k polygonal subdivision of a circle and number the k discrete normal directions $\mathbf{n}$ of the polygon boundary with integers from 1 to k. See also Figure 1.

Now number each line segment of the contour boundary with the integer i if it has the largest dot product of its normal with the i^{th} normal of the regular polygon. Under this mapping the k discrete normal directions on the circle partitions the polygonal contour on a data slice into groups where the members of a group consist of a connected sequence of line segments having the same assigned number. The endpoints of groups are the contour points whose two incident line segments have distinct assigned numbers. At these contour points, normal vectors are assigned as the average of the normals of the incident line segments. The line segments of each group are then replaced by a single conic which

C^1-interpolates the group endpoints and the assigned normal vectors and simultaneously least-squares approximates the contour line segments that originally formed the group. The C^1 interpolation of the pair of endpoints and assigned normal vectors imposes four linear constraints on the coefficients of the conic leaving one degree of freedom for the least-squares approximation within in each group. For details on efficiently computing such C^1 interpolations see [3]. If the least-squares approximation yields a poor error bound then additional conics can be used to achieve a better bound. This operation is of course local to the the group and can be achieved by selectively refining the regular polygon edge corresponding to that group, replacing that edge by two or more edges inscribed in the circular arc subtended by that edge. The newly created normal directions are now mapped to the polygonal contour splitting the group into sub-groups. Each sub-group can now be replaced by a conic, improving the approximation.

Next, using a algorithm similar to the one in [6], the piecewise conic contours on adjacent selected data slices are connected to form a network of triangular facets. The connections between adjacent contours occur exactly at the end points of the conics. Each of these connections is a polynomial cubic curve which C^1 interpolates the end points and the assigned normal vectors, one endpoint each on an adjacent contour. Here conics may not suffice as C^1 connecting curves since the assigned normal vectors at the end points may induce an inflection point. Hence cubic curves are chosen. Notice that this problem does not arise in the planar conic contour generation stage, where a pair of conics can be substituted for such a similar pair of bad normals. See also [4] where necessary and sufficient conditions on a pair of endpoints and assigned normal vectors have been derived for the existence of a proper C^1 interpolating conic.

Furthermore, for each triangular facet consisting of two cubic curves and one conic, specify a quadratic variation of normals along the conic and a cubic variation of normals along the cubic curves. See [4] for details of how this can be efficiently achieved. Finally, each triangular facet of curves and specified normals is C^1 interpolated by an algebraic surface of degree at most seven, yielding a globally C^1 continuous mesh of triangular algebraic surface patches which model the skeletal structure. For details on efficiently computing such C^1 interpolations see [3]. The following Theorem proves why degree seven algebraic surface patches suffice.

Theorem 2.1 *At most $16n-11$ linear constraints have to be satisfied by the coefficients of an algebraic surface of degree n which C^1 interpolates a triangle consisting of two cubic curves and one conic.*

Proof: To C^1 interpolate a conic curve requires $4n$ linear constraints, and to C^1 interpolate a cubic curve requires $(6n-1)$ linear constraints to be satisfied by the coefficients of an algebraic surface of degree n [3]. For the triangle of two cubic curves and one conic, there are totally $4n+2(6n-1) = 16n-2$ independent constraints to be satisfied. However since the triangle vertex points and the specified normals at those points are common between the curves, there are 9 dependencies amongst these constraints. Hence the total number of independent linear constraints are no more than $16n-11$. ♣

The minimum degree of the interpolating algebraic surface is 7, since there are 120 coefficients (119 degrees of freedom) and the independent linear constraints are no more than 101, which results in a family of interpolating surfaces with at least 18 degrees of freedom in selecting an instance surface from the family. These degrees of freedom are chosen, (a) to least squares approximate the points on the polygonal contours lying intermediate to the earlier selected data slices, and (b) to obtain a single-sheeted surface patch.

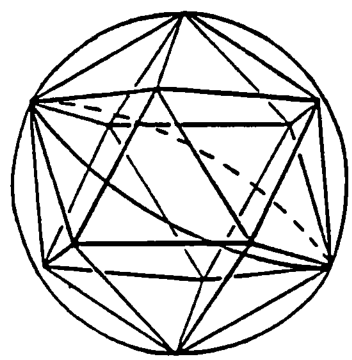

Figure 2: A regular subdivision of the space of normals on a closed solid object

3 Working Directly in Voxel Space

Our first step is to compute a triangular approximation $\mathcal{T}$ of the surface of the entire skeletal object in three dimensional space from the CT/NMR voxel data. Any of the algorithms, [6] which works on an optimal triangular tiling of contours or [9] which works directly in voxel space, can be used for this purpose. The modeled skeletal object is closed, enclosing finite volume, and so is the triangular approximation $\mathcal{T}$.

Next we compute a compact C^1-continuous piecewise surface approximation of this closed triangulation, where the large number of triangles of $\mathcal{T}$ are replaced by a small number of curved surface patches. The placement of the triangular algebraic surface patches is computed by a generalization of the curvature adaptive scheme of the previous section to three dimensions. The points on a unit sphere are in one-to-one correspondence with the normal directions of the triangular faces which make up $\mathcal{T}$ in three dimensions. Consider a regular icosahedron subdivision of a sphere. Number the 20 discrete normal directions $\mathbf{n}$ of the icosahedron boundary with integers from 1 to 20. See also Figure 2.

Now number each triangular face of $\mathcal{T}$ with the integer i if it has the smallest dot product of its normal with the i^{th} normal of the regular icosahedron. Under this mapping the 20 discrete normal directions on the sphere partitions $\mathcal{T}$ into several groups where the members of a group consist of a connected set of triangular faces having the same assigned number. The boundary separating the groups are the sequence of edges of $\mathcal{T}$ whose two incident triangular faces have distinct assigned numbers.

Next, we replace each group by a much smaller approximating set of fat triangles in a manner that still preserves a closed triangulation i.e. there are no holes in the new triangulation $\mathcal{T}_1$ which consists of these fat triangles. Consider the vertices of $\mathcal{T}$ which have three or more incident triangular faces with distinct assigned numbers. These vertices partition the boundary edges of a group into polygonal chains in space, such that each polygonal chain is common to exactly two adjacent groups.

Each polygonal chain is replaced by a much smaller approximating set of line segments, preserving the end vertices of the polygonal chain. See [7] where both heuristic and optimal algorithms are discussed for achieving such linear approximations. After all polygonal chains of $\mathcal{T}$ are processed in this manner we are left with a much smaller number of vertices and edges bounding the original groups of $\mathcal{T}$. These vertices and edges are part of the new triangulation $\mathcal{T}_1$. Each new group is now separately triangulated by choosing non-intersecting diagonal edges joining vertices on the boundary of that group. The triangulation of the groups yield all the fat triangles of the new triangulation $\mathcal{T}_1$.

Now, replace each face of the new triangulation $\mathcal{T}_1$ by a curved surface patch in the following manner. At the vertex points of $\mathcal{T}_1$, normal vectors are assigned as the average of the normals of the incident triangular faces. This provides a particular tangent plane for all patches which shall C^1 interpolate that vertex. Next, construct a curvilinear wire frame by replacing each edge of $\mathcal{T}_1$ with a cubic curve which C^1 interpolates the end points of the edge and the specified "normals". Any remaining degrees of freedom of the C^1 interpolatory curve are used to least-squares approximate all the orginal vertices of $\mathcal{T}$ which were replaced by the single edge of $\mathcal{T}_1$. See [3] for details of how this can be efficiently achieved. Specify a cubic variation of normal vectors along each of the new cubic edge curves. See [4] for details of how this can be efficiently achieved. This specifies the tangent planes for the two incident patches which shall C^1 interpolate the edge curves. Finally, C^1 interpolate the three edge curves and curve normals of each face by a degree seven algebraic surface. The following Theorem proves why degree seven algebraic surface patches suffice.

Theorem 3.1 *At most $18n-12$ linear constraints have to be satisfied by the coefficients of an algebraic surface of degree n which C^1 interpolates a triangle consisting of three cubic curves.*

Proof: To C^1 interpolate a cubic curve requires $(6n-1)$ linear constraints to be satisfied by the coefficients of an algebraic surface of degree n [3]. For the triangle of cubic curves, there are thus totally $3(6n-1) = 18n-3$ independent constraints to be satisfied. However since the triangle vertex points and the specified normals at those points are common between the curves, there are 9 dependencies amongst these constraints. Hence the total number of independent linear constraints are no more than $18n-12$. ♣

The minimum degree of the interpolating algebraic surface is 7, since there are 120 coefficients (119 degrees of freedom) and the independent linear constraints are no more than 114, which results in a family of interpolating surfaces with at least 5 degrees of freedom in selecting an instance surface from the family. These degrees of freedom for each individual patch are chosen, (a) to least square approximate all the triangles in the original group of $\mathcal{T}$, and (b) to obtain a single-sheeted surface patch. The resulting surface patches yield a globally C^1 smooth curved model of the skeletal object whose triangular approximation $\mathcal{T}$ was constructed from the voxel data.

4 Implementation

All the above algorithms have been implemented in SHASTRA, a distributed and collaborative design environment [1]. The SHASTRA environment currently consists of three indpendent toolkit processes, VAIDAK, SHILP and GANITH, which communicate and share data structures and algorithms via UNIX inter process communication. The planar surface reconstruction from image data is performed in the VAIDAK medical imaging and model reconstruction toolkit. The smooth surface

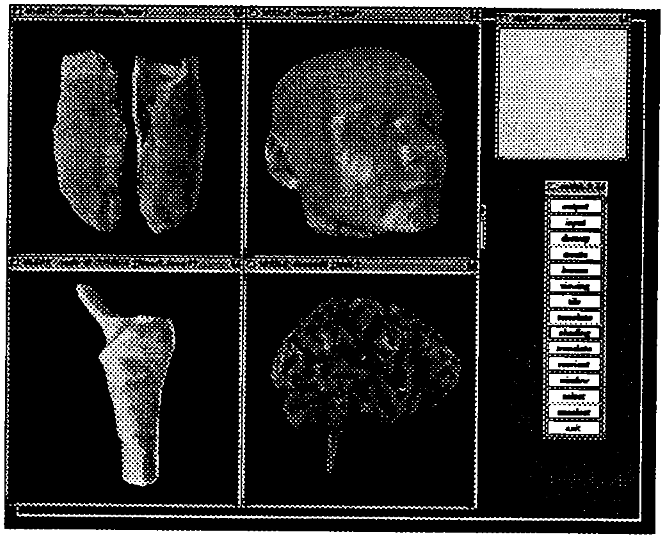

Figure 3: Smooth Models of Parts of the Human Anatomy

fitting is accomplished in the SHILP solid modeling and display toolkit which additionally makes remote procedure calls to the GANITH polynomial and power series manipulation toolkit, for the curve and surface C^1 interpolation and least-squares approximations.

Acknowledgments Thanks to Brian Bailey and Dan Schikore, two very bright undergraduate students, for their assistance in the implementation of the above algorithms.

References

[1] V. Anupam, C. Bajaj, and A. Royappa. *The SHASTRA Distributed and Collaborative Geometric Design Environment.* Computer Science Technical Report, CAPO-91-38, Purdue University, 1991.

[2] C. Bajaj. Geometric modeling with algebraic surfaces. In D. Handscomb, editor, *The Mathematics of Surfaces III*, pages 3–48. Oxford Univ. Press, 1988.

[3] C. Bajaj and I. Ihm. Algebraic surface design with Hermite interpolation. ACM Transactions on Graphics, 1991.

[4] C. Bajaj and I. Ihm. *C^1 Smoothing of Polyhedra with Implicit Surface Patches.* Computer Science Technical Report, CAPO-91-27, Purdue University, 1991.

[5] L. Brevdo, S. Sideman, and R. Beyar. A Simple Approach to the Problem of 3D Reconstruction. *Computer Vision, Graphics and Image Processing*, 37:420–427, 1987.

[6] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal Surface Reconstruction from Planar Contours. *Communications of the ACM*, 20:693–702, 1977.

[7] I. Ihm and B. Naylor. Piecewise linear approximations of digitized space curves with applications. In N.M. Patrikalakis, editor, *Scientific Visualization of Physical Phenomena*, pages 545–569. Springer-Verlag, Tokyo, 1991.

[8] E. Keppel. Approximating Complex Surfaces by Triangulation of Contour Lines. *IBM J. Research and Development*, 19:2–11, 1975.

[9] W. Lorensen and H. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics*, 21:163–169, 1987.

[10] R. Gmelig Meyling. Least Squares B-spline Surface Reconstruction in Tomography. *Manuscript*, 1984.

[11] R. Gmelig Meyling and P. Pfluger. B-spline Approximation of a Closed Surface. *IMA J. of Numerical Analysis*, 7:73–96, 1987.

[12] D. Moore and J. Warren. Approximation of dense scattered data using algebraic surfaces. In *Proc. of the 24th Hawaii Intl. Conference on System Sciences*, pages 681–690, Kauai, Hawaii, 1991.

[13] G. Nielson and T. Foley. Knot Selection for Parametric Spline Interpolation. In T. Lyche and L. Schumaker, editors, *Mathematical Methods in Computer Aided Geometric Design*, pages 261–271. Academic Press, 1989.

[14] V. Pratt. Techniques for conic splines. *Computer Graphics*, 19(3):151–159, 1985.

[15] L. Schumaker. Reconstructing 3D Objects from Cross Sections. In W. Dahmen, M. Gasca, and C. Michelli, editors, *Computation of Curves and Surfaces*, pages 275–309. Kluwer Academic Publishers, 1990.