Skeletal Reconstruction of Branching Shapes

Eric Ferley, Marie-Paule Cani, Dominique Attali

Skeletal Reconstruction of Branching Shapes

Eric Ferley †, Marie-Paule Cani Gascuel †, Dominique Attali ‡

† iMAGIS *- GRAVIR / IMAG
BP 53, F-38041 Grenoble cedex 09, France
Eric.Ferley@imag.fr, Marie-Paule.Gascuel@imag.fr

‡ Infodis - TIMC / IMAG
IAB, Domaine de la Merci, 38706 La Tronche cedex, France
Dominique.Attali@imag.fr

Abstract

We present a new method for the implicit reconstruction of branching shapes from a set of scattered data points. The method is based on the computation of a geometric skeleton inside the data set. This skeleton is simplified in order to filter noise and converted into skeletal elements – a graph of interconnected curves – that generate an implicit surface. We use Bézier triangles as extra skeletal elements to perform bulge free blends between branches while controlling the blend extent. This leads to a smooth implicit representation of the shape, directly computed in a purely geometric way.

1 Introduction

With the recent developement of advanced range-imaging sensors, automatic reconstruction of real-world objects from scattered data points has become an important issue in Computer Graphics. Applications include medical imaging, inverse engineering, simulation, and semi-automatic modeling for design or animation. For the first set of applications, the aim is to provide an improved and enhanced visualization of the data. In other cases, the reconstructed object is going to be edited, deformed and animated. Here, providing a structured reconstruction is essential.

In the area of shape recognition, researchers structure a data set by computing its *skeleton*, defined as the locus of the centers of maximal spheres inside the data [Blu67, ASdB93]. This skeleton, a thin centered structure that throughout this paper will be called *geometric skeleton*, provides a compact representation of both the topology and the geometry of the shape. Such a notion of a geometric skeleton appears attractive in the area of implicit surface modeling. The questions to be asked are: dœs this skeleton represent the same notion as the

*iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.

skeletal elements used to generate an implicit surface [BW90, BS91, Blo95a]? Can we use it to construct a smooth implicit representation of an object?

Compared with previous approaches for reconstruction using implicit surfaces, the introduction of a geometric skeleton to generate the surface gives important benefits:

- The implicit surface can be computed in a purely geometric way, without the need of optimization processes such as in [Mur91, BTG95]. These processes are often computationally intensive and difficult to control.
- The skeleton, a graph of interconnected curve segments or surface elements, defines a structure for the reconstructed object. It can be used to edit the shape in an intuitive way. Such a structure is not provided when objects are reconstructed by directly computing an implicit function [Whi95, VG95], or by merely generating a list of skeletal points [Mur91, BTG95].

In consequence, an approach based on skeletal reconstruction seems more adapted to modeling and animation applications.

This paper focusses on the reconstruction of free form branching shapes from the geometric skeleton of data points scattered on the object's surface. Conversion from geometric skeleton to boundary representation has already been widely studied for parametric surfaces [GD95]. The aim of this paper is different since we want to build a smooth implicit surface representing the shape. We call "branching shapes" these objects for which the geometric skeleton can be represented as a graph of interconnected curves. Extensions to skeletons that include surface elements are discussed in the conclusion.

Section 2 reviews the definition of the geometric skeleton of an object and presents the semi-continuous approach that was used to compute it from a data set. Section 3 points out that this skeleton cannot directly be used as an input to generate an implicit surface, and lists the problems to solve. Section 4 presents a method to convert each branch of the geometric skeleton into a skeletal curve that generates a smooth implicit surface of varying radius in the cross-section. Section 5 develops a new method, based on Bézier triangular skeletal elements, to smoothly blend branches at joints without generating bulges or creases. Results are presented in Section 6. We conclude and focus on future work in Section 7.

2 Geometric Skeleton

2.1 Definitions

The **geometric skeleton** of an object is the locus of the centers of maximal spheres inside this object. A sphere included in an object is said to be **maximal** if it is not included in any other sphere included in the object.

The geometric skeleton is a thin structure, represented as a graph of interconnected curve segments or surface elements, centered in the object (see Figure 1). It stores in a compact way both the topological properties of the

object – given by its graph structure – and the geometrical information, i.e. the distance to the surface.

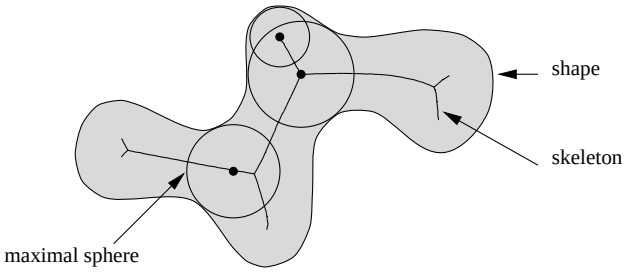


Figure 1: The geometric skeleton of an object.

Note that the geometric skeleton is based on the same principles as the "medial axis" used in discrete geometry [LLS92]. However, the medial axis of a data set is built as a volume made of an unstructured set of voxels.

In contrast, the graph structure of the geometric skeleton gives useful neighboring information. The next section explains how the geometric skeleton, which may initially include surface elements, is approximated by a set of polylines.

2.2 Computation of the geometric skeleton

The geometric skeleton of an object is computed using the continuous approach described in [BA92]. The input is a set of points $\{p_i\}_{i=1}^{n}$ located on the surface of the object. The output is a subgraph of the Voronoi graph of the boundary points p_i.

The Voronoi graph is a well known data structure in computational geometry [Aur91]. In summary, the Voronoi graph of a set of points (called seeds) divides the space into regions. Each region is the set of points closer to a particular seed than to any other seed.

In [BA92], the geometric skeleton is approximated with the Voronoi elements completely included in the object (see figure 2).

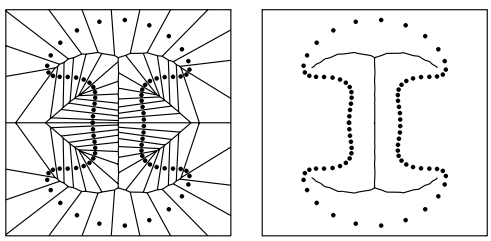


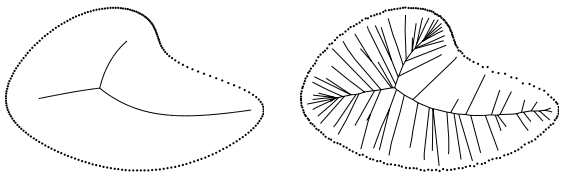
Figure 2: Relation between the Voronoi graph (a) and the geometric skeleton (b) of a set of point.

The result is a thin shape made up of straight-line segments in $2D$ and polygons in $3D$ (Figures 3(a) and 4(b)). Each Voronoi vertex located on the

geometric skeleton is the center of a sphere passing through at least three seeds and having no seed in its interior. The radius of this sphere approximates the distance to the surface at each vertex.

2.3 Pruning branches due to noise

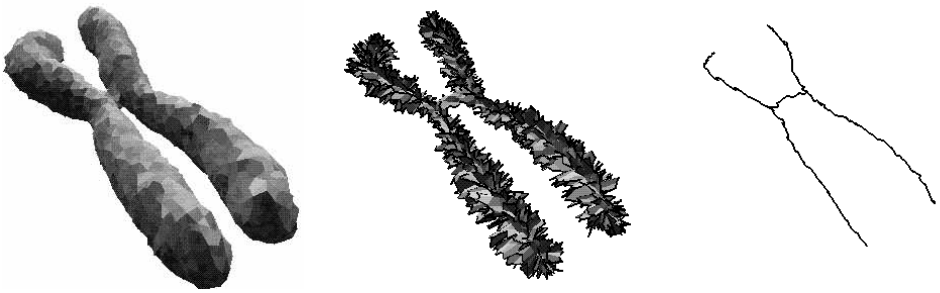
A drawback of the skeleton transformation is its sensibility to noise. Noise on the boundary of an object may significantly change the aspect of its skeleton (Figures 3(b)). A simplification algorithm is therefore necessary to remove peripheral branches having no perceptual relevance.



(a) (b)

Figure 3: Sensitivity of the geometric skeleton to noise in the data.

Attali and Montanvert [AM94] propose a method of simplification of continuous skeletons, that are made of both line segments and surface elements in the general case (see Figure 4). Peripheral branches and polygons are removed one after the other while they satisfy a *removing criterion*. By construction, the simplified skeleton is a subset of the initial skeleton that has the same class of homotopy. In $3D$, the result is a **wireframe figure**. The order of all branching points is three.



(a) (b) (c)

Figure 4: The data set of a chromosome, its geometric skeleton (including surface elements), and the wireframe skeleton obtained at the end of the simplification process.

3 Problems Raised by Skeletal Implicit Modelling

For people working in the area of implicit modeling, the use of a skeleton – ie. a set of geometric primitives admitting well-defined distance functions – to generate an implicit surface is an intuitive concept. Ideally, a smooth surface

representing an object should be directly computed from any skeletal description.

Unfortunately, using the geometric skeleton described in Section 2 for iso-surface generation is not trivial. This section first reviews methods for skeletal implicit modeling, and then points out the difficulties raised by the use of the geometric skeleton computed in Section 2.

3.1 Implicit surfaces generated by skeletons

Implicit surfaces have been defined as iso-surfaces of field functions[1] generated by a set of geometric primitives that define the "skeleton" of the object [BW90, BS91].

The implicit contribution f_i of each skeletal primitive S_i is a decreasing function of the distance to this element. The implicit surface $\mathcal{S}$ is defined as an iso-surface of isovalue c for the sum of skeletal contributions:

$$\mathcal{S} = \left\{ P / \quad \sum_i f_i(P) - c = 0 \right\} \tag{1}$$

This formulation is a generalization of the original Blinn's objects [Bli82], metaballs [NHK+85] and soft objects [WMW86], where only skeletal points were used.

If the summation is replaced by a maximum in equation (1), the implicit volume is the union of the volumes created by each skeletal primitive (see Figure 5(c)). The advantage of summation is that it defines a smooth blend between these contributions.

However, the way each f_i varies affects the blend and generating a smooth shape is not easy. In particular, bulges (see figure 5(b)), characterized by a *"cross section that exhibits negative, then positive, then negative curvature with respect to the underlying skeleton"*[2], should be avoided.

The next paragraphs review the different models that have been proposed for field functions.

Distance surfaces

Distance surfaces [BW90] define f_i as a decreasing C^1 continuous function F of the distance[3] to S_i:

$$f_i(P) = F(d(P, S_i)) \tag{2}$$

The choice of a function F with local support limits the influence of each skeletal component, thus providing local control of the shape and reducing computations. Anisotropic field functions depending on the location of the closest point on S_i may be used to generate more complex shapes [KAW91].

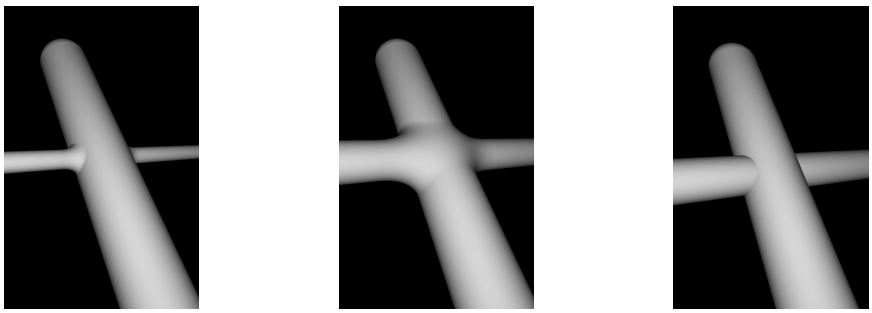
Distance surfaces can only be used for convex skeletons, otherwise creases will appear. Dividing non-convex skeletons into convex parts is not a solution,

[1] Sometimes called "implicit function" although they are defined by an explicit equation.

[2] This definition was proposed by Jules Bloomenthal in [Blo95a].

[3] Defined as the distance to the closest point on S_i

since bulges at joints would be produced in this case. Consequently, modeling a ramification without any bulge or crease is difficult (see Figure 5).



(a) (b) (c)

Figure 5: Distance surface blend for a skeleton made of two segments:
(a) The blend is smooth when the thicknesses are very different.
(b) A bulge appears when the branches have about the same thickness.
(c) Union surface produced by replacing sum by maximum in equation (1).

Procedural methods

Procedural field functions were introduced in [BW90] as an example of complex implicit definition. For a specific example of two branching curves S_1 and S_2, the two points P_1 and P_2 respectively closest to P on the two skeletal curves S_i are computed, and then $f_i(P)$ is defined by the distance to the closest point on the segment $[P_1, P_2]$:

$$f_i(P) = F(d(P, [P_1, P_2]))$$

Unfortunately, this approach that seems to offer an elegant solution to the bulge problem cannot be applied in the general case:

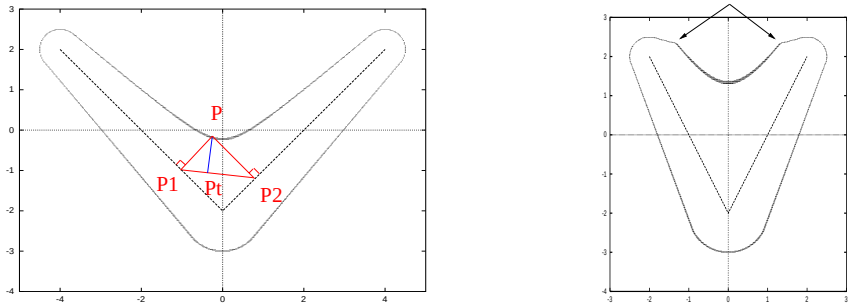
- It dœs not work for sharp angles between branches, as shown in Figure 6, since the surface tangent changes suddenly when the projected point P_1 (respectively P_2) is located on the extremity of the line segment.
- There is no control of the blend extent: it is restricted by the projective domains of points P on S_i.

Two consecutive ramifications cannot be modeled with this method.

Convolution surfaces

Convolution surfaces [BS91] define f_i as the integral over S_i of contributions from individual points. Here, the resulting surface dœs not depend on the way a skeleton is divided into skeletal elements, which is more satisfactory. Moreover, the resulting surface is smooth, even for a non-convex skeleton.

Convolution surfaces solve the bulge problem if certain constraints are applied to the skeleton. Previously, these constraints have been applied to skeletons consisting of surfaces or volumes. No solution has been proposed in the



(a) (b)

Figure 6: Effects of the procedural method to blend the contributions of two line segments:
(a) The method works when the angle between the segments is greater than $\frac{\pi}{2}$.
(b) Strange features appear for sharp angles.

case of one dimensional primitives: a bulge will appear near the joint in the 3D case. The solution that uses a thick volume skeleton instead of curves is not applicable in the general case, since convolution of generalized cylinders would be difficult to compute.

3.2 Problems to be solved

Applying one of the previous skeletal modeling techniques to reconstruct a smooth shape from the geometric skeleton of a data set is not straightforward:

1. Each branch of the geometric skeleton computed in Section 2 is made up of many segments. Due to noise in the data, the resulting polyline may not be very smooth. The distance surface generated by this polyline would exhibit creases, while the summation of each segment's contribution would generate bulges. The use of convolution surfaces would be computationally intensive regarding the number and size of the segments.

2. In order to create a smooth reconstructed surface, we must be able to define a smooth blend between connected branches of the skeleton. None of the previously mentioned implicit techniques gives a direct answer to this problem.

Section 4 describes a solution to the first problem, based on the conversion of skeleton branches into smooth skeletal curves.

Section 5 presents a new method to perform a bulge-free blend of the implicit surfaces generated by branching curves.

4 Implicit Reconstruction along a Branch

4.1 Polyline Simplification

The continuous approach of Section 2 produces a skeleton whose number of vertices is of the same order of magnitude as the number of the data points.

Consequently, each branch of the geometric skeleton is an irregular polyline made up of a large number of very small segments (see Figure 4).

A first step before using these polylines in skeletal implicit modeling is to reduce their number of vertices, according to the local smoothness of the polyline and the desired level of detail.

We use the recursive algorithm of Pavlidis [PH74]: each polyline is first approximated by the line segment between its extremities, and then this coarse representation is recursively refined by adding some of the inner initial vertices. The algorithm to refine a segment is:

1. Compute the distance between the segment and each of the initial vertices along this branch;
2. Stop the refinement if the maximal distance is smaller than the level of detail threshold;
3. Otherwise, split the segment into two parts by adding the most distant of the initial vertices, and recursively apply the same process to the two resulting segments.

For our application, we need to take into account the variations of the object's thickness along a polyline to decide how much it should be simplified. Thus, we use the above algorithm in a 4D space, where the last coordinate of each skeleton vertex is the distance to the surface (given by the radius of the associated maximal sphere).

4.2 Conversion to a smooth curve

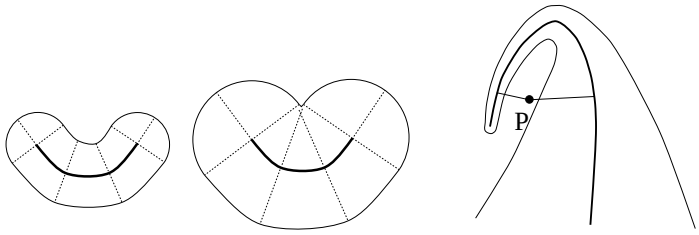
Polylines computed in the previous section cannot directly be used as skeletal elements generating a distance surface, since creases would appear in concavities. In order to get a smooth distance surface, we smooth the branches of the skeleton by converting the polyline into curves.

At this point, we may remark that the distance surface generated by a curve is smooth provided that the thickness remains locally smaller than the local radius of curvature (see Figure 7(a)). Under the hypothesis that we are trying to reconstruct a smooth shape, this property should always be satisfied.

Considering this remark, we approximate the polyline by a cubic B-spline curve, whose control points are the vertices of the simplified polyline. At the moment, if the curvature property is not verified, we increase the polyline simplification.

4.3 Implicit contribution along a curve

The skeleton gives precise information on the geometry of the object, since the distance to the surface is stored for each vertex. The reconstructed implicit surface should take this information into account. In particular, using distance to the closest point on the curve to define the implicit contribution at point P, as in equation (2), no longer works (see Figure 7(b)). Euclidean distance must



(a) (b)

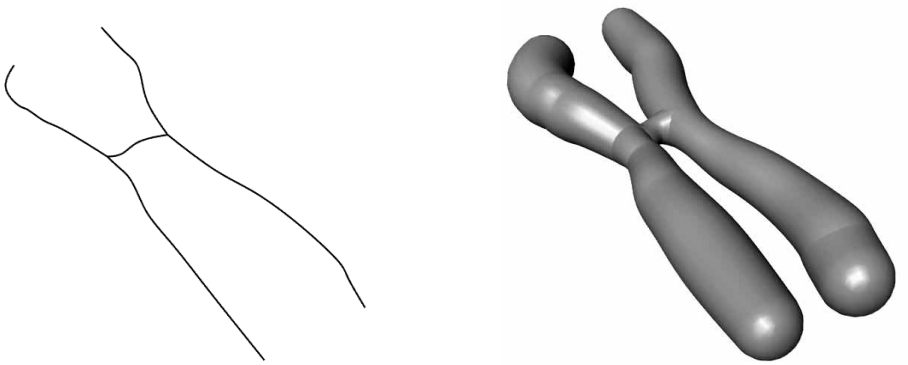
Figure 7: Implicit contribution along a curve.

be replaced by a generalized distance that takes the radius along the curve into account.

The implicit contribution $f_i(P)$ of varying strength along a cubic curve $S(u)$ can be for instance computed as in [Blo95b], by computing the zeros of a five degree polynomial, and taking the one that minimizes the distance normalized by the thickness.

In practice, our figures were generated by another method inspired by the parametric approach of [GD95].

Figure 8 shows the conversion of the geometric skeleton of the chromosome into a graph of B-spline curves, and the union surface obtained by using each of these curves in isolation to generate an implicit surface of varying radius.



(a) (b)

Figure 8: Implicit reconstruction along branches for the chromosome example: (a) Conversion of the polylines into cubic B-spline curves of 51 control points. (b) Union surface composed of implicit surfaces of varying radius generated along each curve.

5 Bulge free blend between implicit branches

The last step of the implicit reconstruction algorithm is to generate a smooth blend between implicit contributions of connected skeletal curves.

5.1 Branches of very different thicknesses

As noticed in Section 3.1, the direct use of blended distance surfaces gives good results when the thicknesses of connected branches are very different (see Figure 5). In this case, the locality of the blend can be controlled by carefully choosing the support of the function F used for each branch.

However, if branches have approximately the same size, a bulge will appear near the junction. As explained in Section 3, none of the previous implicit modeling techniques gives a solution to this specific problem in the case of one dimensional skeletal primitives.

The next sections present a new solution to bulge free blending, based on the local thickening of skeletons at joints. The method has already been implemented for joints between three skeletal curves. We are currently extending it to more general junctions.

5.2 Blending three branches

Our solution proposes to smooth the joint by locally replacing the three curves by a small surface patch. Then, the union of the curves and this patch, which is a smooth non-manifold entity, will be considered as a single skeletal element from which distance surfaces will be generated.

In the case of three skeletal curves, we use a cubic Bézier triangle [Far88] to smooth the joint. This triangle, defined by 10 control points, is represented by the parametric equation:

$$\begin{aligned} S(r,s,t) &= r^3P_0 + s^3P_1 + t^3P_2 + 6rstP_9 \\ &+ 3r^2sP_3 + 3rs^2P_4 + 3s^2tP_5 + 3st^2P_6 + 3rt^2P_7 + 3r^2tP_8 \end{aligned}$$

Points P_0, P_1 and P_2 are interpolated, while points P_3 to P_8 define the tangent

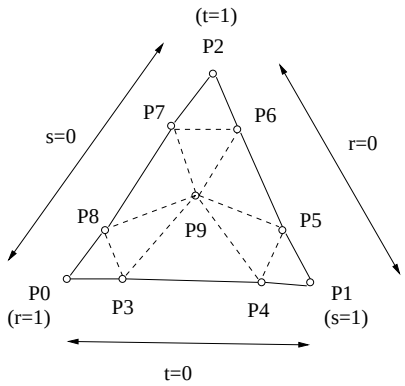


Figure 9: Definition of a Bézier triangular patch.

directions (see Figure 9).

The points where the triangle is attached to the curves control the blend extent. We choose them according to the thickness of the solid at the joint. The same property as in Section 4.2 must be verified: the radius of curvature of the Bézier triangle must be larger than the surface radius, otherwise no smooth

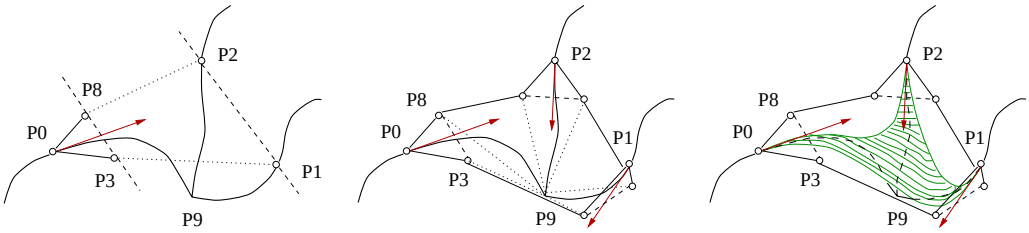


Figure 10: Definition of an extra-skeletal element.

distance surface can be defined.

This restrained us from merging the tangents to the triangle at its extremities (by merging P_3 and P_8, P_4 and P_5 and P_6 and P_7): that would have ensured the G^1-continuity between the curves and the triangular patch, but then the radius of curvature would have reached zero near their connexion.

So, we proceed in two steps:

- First, we position a triangular patch satisfying the curvature condition (see figure 10(a) and (b)).
- Then, we restrict the patch to ensure the G^1-continuity between the curves and the patch. This is done by composition of Bézier triangles: a 2D triangle defines the parameter space of the previously positioned triangle (see figure 10(c)).

Implicit contribution along the Bézier triangle

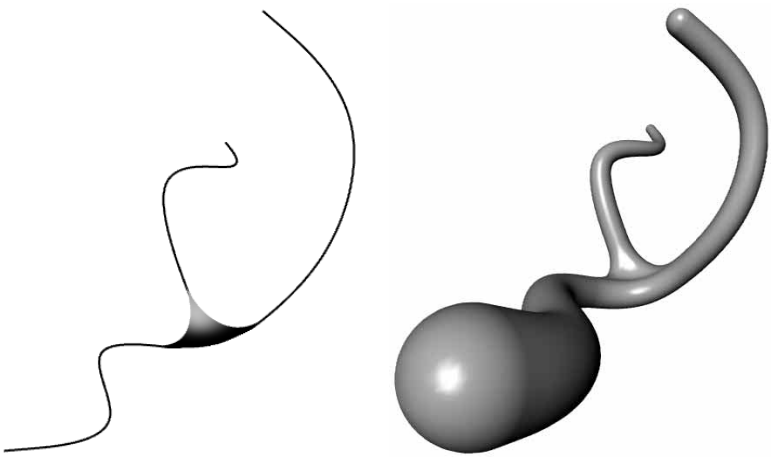
When the surface has a varying radius along the three skeletal curves, the distance to the implicit surface must vary smoothly along the Bézier triangle $\mathcal{T}$. We use the same approach as in Section 4.3: the equation of the Bézier triangle is used again to interpolate the distances to the surface at control points, in order to compute the desired radius $r(r, s, t)$ at the point of barycentric coordinates r, s, t in $\mathcal{T}$. Then, the implicit contribution along its surface can be computed in the same way than along a curve (section 4.3). Again, our implementation is based on an explicit method to display the corresponding *offset* surface (inspired from [GD95]).

Figure 11 shows the results for 3D curves along which the thickness is varying.

5.3 Generalizations

Although not yet implemented, extensions of this blending approach to more general branchings seem promising.

In the case of a flat branching between n curves, the method would first project the curves to a plane that is "tangent" to the joint, in order to define



(a) (b)

Figure 11: Bulge free blend between three skeletal curves

an order between the curves. Then, an n-sided Bézier patch [LD89] could be used instead of a Bézier triangle to thicken the junction.

In the general branching case, the n branches may go in any direction in space, so there is no way of flattening the junction. Then, the generalization of our method would mean embedding the joint in a Bézier volume interpolating each of the n branches (see figure 12).

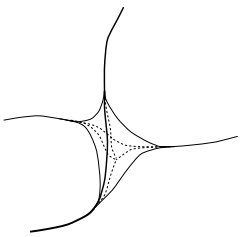
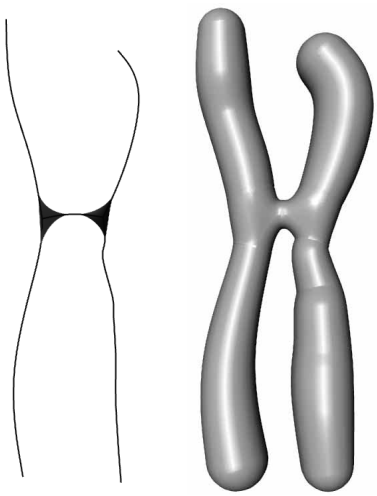


Figure 12: Embedding of a joint in a Bézier volume

The faces of this Bézier volume would be Bézier triangles joining three of the n curves, so computing the closest point of this volume would involve closest point computations for Bézier triangles only.

6 Results

Figure 13 shows the final reconstruction for the chromosome example of Figure 4. Figure 14 illustrates the different steps of the reconstruction of a more complex object.



(a) (b)

Figure 13: Smooth implicit reconstruction of the chromosome.

7 Conclusion and Future Work

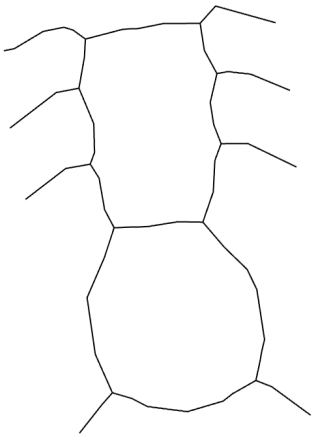
This paper has developed a new approach to the reconstruction of 3D free-form solids, based on the combination of geometric skeletons and implicit surfaces. The main features of the method are:

- a purely geometric way to perform the reconstruction, without any optimization process to define the skeletal elements that generate the surface.
- a structured reconstruction: the skeleton that generates the implicit surface is a graph of interconnected curves, so it gives an intuitive and compact representation of the object. Editing this skeleton to model deformations or articulations appears also feasible. Therefore, the method is potentially adapted to further applications in modeling and animation.

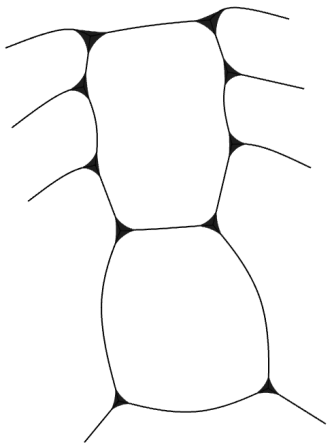
Perfecting the reconstruction method has led us to solve a problem inherent in skeletal implicit design: the bulge free blending of surfaces generated by skeletal curves. Instead of trying to define more complex implicit contributions, as was done with convolution surfaces or with procedural field functions, our solution is based on the local thickening of the skeleton near a joint to prevent the creases on the corresponding distance surface. This approach offers the advantage of an explicit control on the blend extent near the joint. In a modeling system, the surface or volume element used to thicken the skeleton could be computed in an automatic way, thus being transparent to the user.

Future work includes the implementation of a smooth blend between any number of skeletal curves, using Bézier volumes. We are also looking for bounds on the curvature of Bézier triangles in order to fix their size automatically according to the local thickness of the object at a joint.

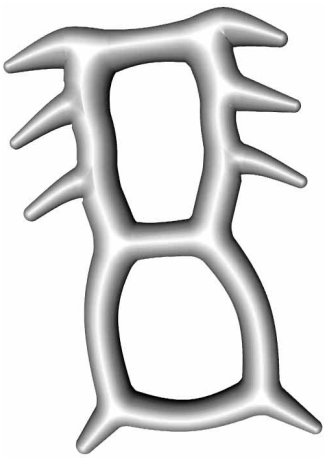
This paper has studied the reconstruction of "branching shapes" for which the skeleton was a graph of interconnected curve segments. In the general case



(a) initial skeleton



(b) final skeletal elements



(c) implicit surface

Figure 14: Reconstruction of a free form branching shape.

the geometric skeleton of a solid includes both curves and surface elements. We are planning to use either interconnected surface patches or curve segments with anisotropic implicit contributions to model these surface elements.

Acknowledgements

Our understanding of the bulge free blend problem was really deepened by heated debates with Mathieu Desbrun, Agata Opalach and Jules Bloomenthal. Many thanks to Mathieu for his help during the implementation and to Agata for carefully rereading this paper.

References

[AM94] D. Attali and A. Montanvert. Semicontinuous skeletons of 2D and 3D shapes. In C. Arcelli et al., editors, *Aspects of Visual Form Processing*, pages 32–41. World Scientific, Singapore, 1994.

[ASdB93] C. Arcelli and G. Sanniti di Baja. Euclidean skeleton via centre-of-maximal-disc extraction. *Image and Vision Computing*, 11(3):163–173, April 1993.

[Aur91] F. Aurenhammer. Voronoi diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 33(3):345–405, 1991.

[BA92] J. W. Brandt and V. R. Algazi. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55(3):329–337, 1992.

[Bli82] J. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, pages 235–256, July 1982.

[Blo95a] Jules Bloomenthal. Bulge elimination in implicit surface blend. In *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, pages 7–20, Grenoble, France, April 1995.

[Blo95b] Jules Bloomenthal. Skeletal design of natural forms. *PHD Thesis*, The University of Calgary, January 1995.

[Blu67] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380, Cambridge, MA, 1967. M.I.T. Press.

[BS91] Jules Bloomenthal and Ken Shoemake. Convolution surfaces. *Computer Graphics*, 25(4):251–256, July 1991. Proceedings of SIGGRAPH'91 (Las Vegas, Nevada, July 1991).

[BTG95] Eric Bittar, Nicolas Tsingos, and M.P. Gascuel. Automatic reconstruction of unstructured 3d data: Combining medial axis and implicit surfaces. In *Eurographics'95*, September 1995.

[BW90] Jules Bloomenthal and Brian Wyvill. Interactive techniques for implicit modeling. *Computer Graphics*, 24(2):109–116, March 1990. Proceedings of Symposium on Interactive 3D Graphics.

[Far88] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, San Diego, California, 1988.

[GD95] S.-M. Gelston and D. Dutta. Boundary surface recovery from skeleton curves and surfaces. *Computer Aided Geometric Design*, 12:27–51, 1995.

[KAW91] Zoran Kacic-Alesic and Brian Wyvill. Controlled blending of procedural implicit surfaces. In *Graphics Interface'91*, pages 236–245, Calgary, Canada, June 1991.

[LD89] C. Loop and T. DeRose. A multisided generalization of bézier surfaces. *ACM Transactions on Graphics*, 8(3):204–234, 1989.

[LLS92] L. Lam, S.-W. Lee, and C. Y. Suen. Thinning methodologies - a comprehensive survey. *IEEEPAMI*, 14(9):869–885, September 1992.

[Mur91] Shigeru Muraki. Volumetric shape description of range data using blobby model. *Computer Graphics*, 25(4):227–235, July 1991.

[NHK+85] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Objects modeling by distribution function and a method of image generation (in japanese). *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, J68-D(4):718–725, 1985.

[PH74] T. Pavlidis and S. L. Horowitz. Segmentation of plane curves. *IEEE trans. in Computers*, 23(8):860–870, 1974.

[VG95] Luiz Velho and Jonas Gomez. Approximate conversion of parametric to implicit surfaces. In *Implicit Surfaces'95—the First Eurographics Workshop on Implicit Surfaces*, pages 77–96, Grenoble, France, April 1995.

[Whi95] Ross Whitaker. Algorithms for implicit deformable models. In *The International Conference of Computer Vision*, Boston, Mass, 1995.

[WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4):227–234, August 1986.