Skeletal Methods of Shape Manipulation

Jules Bloomenthal and Chek Lim
Unchained Geometry Inc.

Abstract

The geometric skeleton is derived from a static object using an implicit 'directions' method; an IK skeleton is derived from and used to manipulate the geometric skeleton. The model may be reconstructed from the modified skeleton using implicit distance and convolution methods.

Introduction

We examine the use of the skeleton to manipulate erstwhile static 3D models. These models may be generated by a design system or a hardware scanner, and may be disseminated through the public domain, catalog services, or scanning services. Originally static, the models may be positioned, oriented, and scaled, but not articulated. They may contain separate parts (such as a car and four wheels), permitting only limited animation.

In order to animate such a model, we first extract its skeleton. Then we articulate the skeleton, from which we reconstruct the animated surface.

The 'IK skeleton' is traditionally used to control an articulable (*i.e.*, nonstatic) model. More recently it has been applied to static models, but this approach has several problems. The skeleton must be created, usually manually; and a correspondence between skeleton and surface must be established. Articulation is limited and arbitrary metamorphosis is not possible.

Geometrically, the 'skeleton' has the precise meaning of medial axis (or surface) of the object. The medial axis is similar to the IK (inverse-kinematic) skeleton, but is (typically) two-dimensional, whereas the IK skeleton is one-dimensional.

The following figure depicts an object, its geometric skeleton, and its IK skeleton. The geometric skeleton consists of 2D surfaces and 1D curves, and is completely surrounded by the object. Every point on the

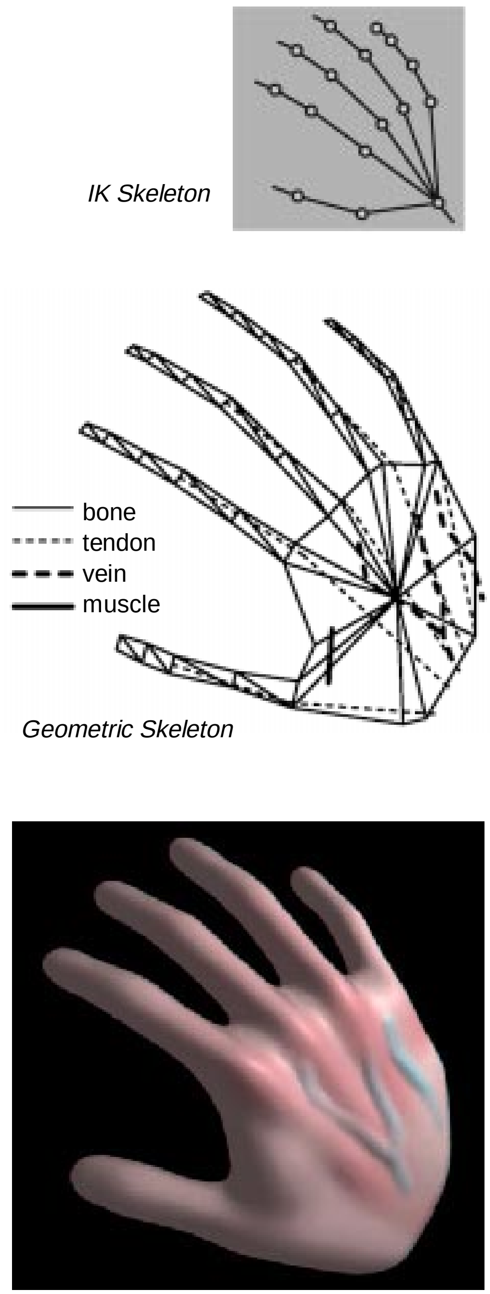

Figure 1. Skeletons and Surface

geometric skeleton is a center of a sphere fully inscribed within the model and touching it at two or more distinct points.

Skeletonization

The geometric skeleton may be derived automatically from the object, and the IK skeleton derived automatically from the geometric skeleton; in the process, the skeleton/surface correspondence is automatically established There appear to be three established methods to extract the geometric skeleton.

Delaunay Triangulation approximates the skeleton as the collection of centers of an optimal triangulation (tetrahedralization) of points on the object's surface; additional points may be needed to provide a sufficiently dense triangulation.

Angle Bisection defines the skeleton as the collection of surfaces internally bisecting the dihedral angles of an original polygonal model; these bisecting surfaces must be carefully trimmed against each other.

Volumetric Thinning is an iterative attrition of voxels originally representing the object's volume; errors accumulate during the iteration.

Because each of these methods has limitations, we developed *Direction Testing*. This is an implicit method that defines the skeleton as the set of points at which the direction to the nearest point on the object undergoes a sudden transition. For example, the central skeletal component for the rectangle below contains a direction change of 180^o; for the sub-components, the change is 90^o.

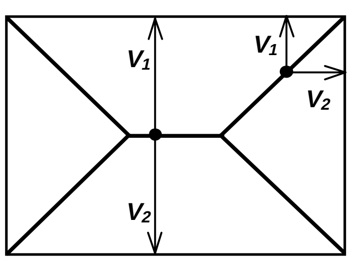

Figure 2. Rectangle, Skeleton, and Directions

Locating and connecting the skeletal points is performed by a piecewise-linear implicit surface polygonizer (see [1] for a survey of polygonization methods). At each lattice point the direction to the nearest point on the object is computed. For those edges connecting points of sufficiently divergent directions, a skeletal point is computed using binary subdivision.

Our polygonizer utilizes adaptive subdivision to improve precision in regions of high curvature, and coalesces coplanar neighboring polygons to reduce storage in regions of low curvature. The lattice is deformed according to [4], eliminating thin or small triangles.

The polygonizer must support the non-manifold and manifold-with-boundary edges that occur in skeletons. An example surface produced by a non-manifold polygonizer is shown below, from [2].

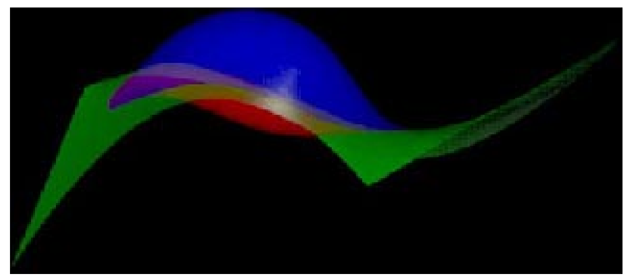

Figure 3. Non-Manifold Polygonization

We optimize the direction computation for a model described by a triangle mesh by assigning to the terminal nodes of an octree the *n* nearest triangles, such that d_n-$d_1 > s$, where *s* is the length of the node's major diagonal and ***c*** is the center of the node. For any point within the octree node, the nearest point on the object must belong to one of the *n* triangles.

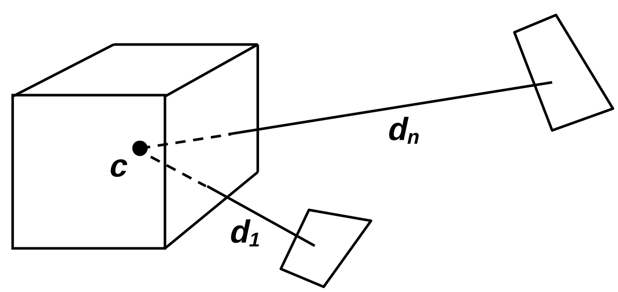

Figure 4. Octree Node and Triangle List

This method appears to offer advantages in speed and precision. The octree is constructed before skeletonization. Assuming its depth is adjusted so that a near constant number of triangles per terminal node is produced, the skeletonization running time is near linear with respect to the geometric extent of the skeleton.

Because a minimum direction divergence is required, small irregularities in the surface do not produce unwanted noise (*i.e.*, branching) in the skeleton (alternatively; this may be considered a disadvantageous loss of surface detail).

Reconstruction

Once the geometric and IK skeletons are extracted from an object, the user manipulates the IK skeleton, which modifies the geometric skeleton. For every point of the geometric skeleton, we associate a distance and angular orientation to a point on the IK skeleton. When the IK skeleton is rotated, twisted, bent, or otherwise manipulated, the geometric skeleton is modified accordingly.

To reconstruct the modified object from the modified geometric skeleton, we implicitly define a distance surface to the skeleton, and polygonize. This rounds those regions of the object that correspond with convex portions of the skeleton. Optionally, we employ convolution surfaces [3] to fillet those regions of the object that correspond with concave portions of the skeleton.

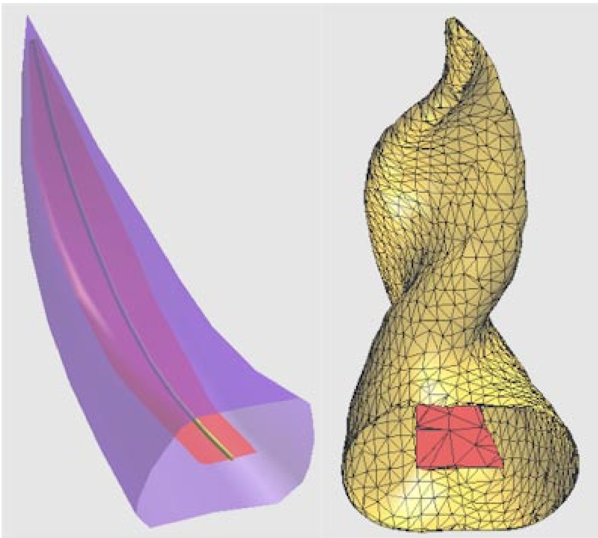

Figure 5. Skeleton and Reconstruction

It is possible to associate points on the object directly with the IK skeleton; some commercial systems, for example, associate an object point with the nearest skeletal joint. This simpler approach can, however, produce self-penetration, shearing, and creasing of the modified object, especially if the modified object is limited to the polygonal mesh of the original object. Some of these artifacts are apparent in the following figure.

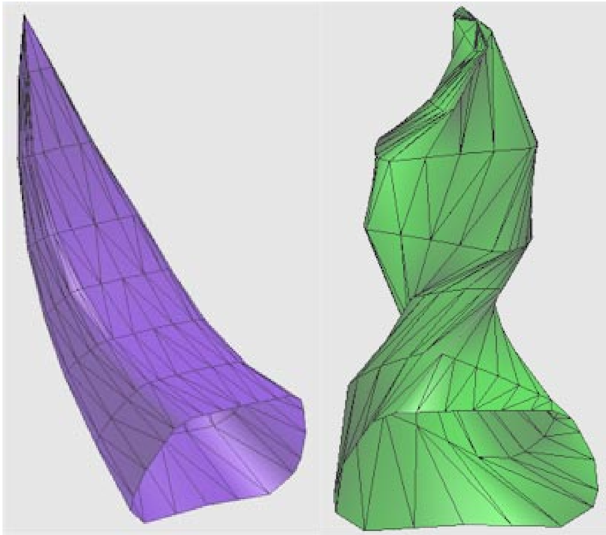

Figure 6. Reconstruction from IK Skeleton

Conclusion and Future Work

Stephen Wainwright once wrote, “structure without function is a corpse, and function without structure is a ghost” [5]. It is the skeleton that underlies the structure and function of an object.

The skeleton also offers an embedding for transformation hierarchies, maps well with motion-capture data, and is easily represented, manipulated, and rendered.

The geometric skeleton (*i.e.*, the medial axis) can be produced automatically from computer models. In conjunction with scanned objects, it permits the animation of otherwise static objects. In conjunction with motion-capture data, it permits the application of real-world motions to sampled or synthesized objects. The geometric skeleton can produce the IK skeleton, obviating a tedious and imprecise manual process.

The methods discussed above are applicable not only to the animation of existing objects, but also to the design of new objects. A skeleton may be constructed interactively or specified procedurally. Skeletons facilitate a constructive and hierarchical approach to modeling and to levels of detail. Two skeletons can be interpolated, permitting the interpolation of two different objects more convincingly than current ‘morph’ methods.

The skeleton is more compact than its corresponding object, requiring less storage and transmission bandwidth. The skeleton supports ancillary properties, such as dynamics.

The skeleton facilitates the design of rounded and filleted surfaces, which are often difficult and tedious to achieve using conventional design tools. CAD systems specify these details with greater precision, however, and have greater application in engineering disciplines. Skeletal design will likely find more application in related disciplines such as industrial design, in which appearance can be more important than geometric precision.

The large design tools provided commercially are unwieldy for the lay person and, even, for many graphic designers. This is due, we believe, to the tools' emphasis on surface manipulation. The use of the skeleton may signal the development of more intuitive design tools.

It would be useful to identify those canonical skeletal operations from which more complex operations can be composed. These could form the basis for clip-behavior, offering a large repertoire of design and animation to the user. Existing shape taxonomies may provide some guidance in the development of these operations.

Is there an optimal language of geometric modeling and, if so, what are its semantics and syntax? What is an optimal form of user interaction? We may ponder these questions for many years, but it may not be long until the skeleton is the established key to desktop 3D.

References

1. Jules Bloomenthal, editor, *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, San Francisco 1997.

2. Jules Bloomenthal and Keith Ferguson, *Polygonization of Non-Manifold Surfaces*, Proceedings of SIGGRAPH 95 (Los Angeles, CA, Aug. 6-11, 1995). In *Computer Graphics* Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 309-316.

3. Jules Bloomenthal and Ken Shoemake, *Convolution Surfaces*, Proceedings of SIGGRAPH 91 (Las Vegas, Nevada, Jul. 28-Aug. 2, 1991). In *Computer Graphics*, 25, 4 (Jul. 1991) ACM SIGGRAPH, New York, 1991, pp. 251-256.

4. Douglas Moore and Joe Warren, *Compact Isocontours from Sampled Data*, Graphics Gems III, David Kirk, editor, Academic Press, New York, 1992].

5. Stephen Wainwright, *Axis and Circumference: the Cylindrical Shape of Plants and Animals*, Harvard University Press, Cambridge, 1988.