

Morphing between Shapes by Using their Straight Skeletons

Gill Barequet Evgeny Yakersberg
Dept. of Computer Science
The Technion—IIT
Haifa 32000, Israel
[barequet,eugeney]@cs.technion.ac.il

Categories and Subject Descriptors

I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Geometric algorithms, languages, and systems*

General Terms

Algorithms, Theory

Keywords

Morphing, straight skeleton, piecewise-linear surface interpolation

1. DESCRIPTION OF THE ALGORITHM

This video segment shows an algorithm that can serve for either interpolating or morphing between planar (polygonal, possibly not simply connected) shapes. In the first application the two planar shapes are assumed to be contained in distinct parallel planes, and the result is a non-self-intersecting surface that connects between the shapes and forms a valid solid object (again, not necessarily simply connected). In the second application the two planes containing the input shapes are assumed to be xy -parallel, and the interpolating surface is cut by a series of intermediate planes. The resulting cut contours form the morphing sequence.

The interpolation algorithm [2] consists of the following steps:

1. First, if this is not part of the input, the hierarchy of contours in each shape is analyzed in order to orient the contours consistently, say, clockwise and counter-clockwise for material and hole contours, respectively.
2. Then, the overlay of the shapes is computed, and the cells of their symmetric difference are identified (Figure 1).
3. Next, the straight skeletons [1] of all the cells identified in Step 2, are computed by using the algorithm of [4]. Each such cell is then triangulated monotonically with respect to its defining contour edge (Figure 2).

Copyright is held by the author/owner.
SoCG'03, June 8–10, 2003, San Diego, California, USA.
ACM 1-58113-663-3/03/0006.

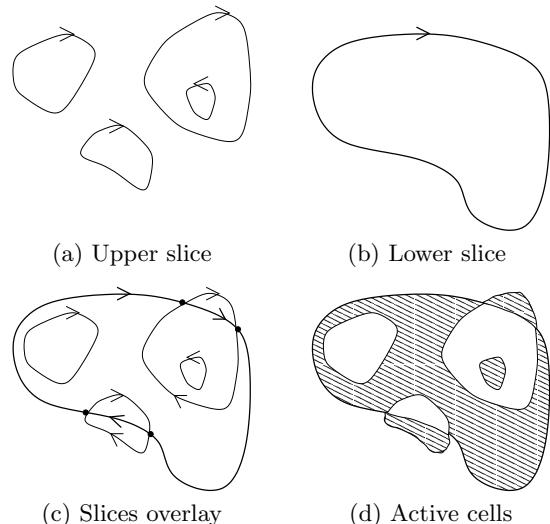


Figure 1: Active cells in the overlay of two slices

4. Finally, all the triangulations of Step 3 are lifted up to space by assigning heights to the skeleton vertices. Original contour vertices are assigned the heights of their respective input planes, while inner skeleton vertices are assigned heights proportional to their offset distance from the contours (Figure 3). This information is “built-in” in the skeleton structures.

The algorithm is somewhat similar to that of Oliva et al. [5] (with some improvements suggested in [4]).

The morphing variant of the algorithm needs only slide the offset polygons between the input shapes, such that vertices

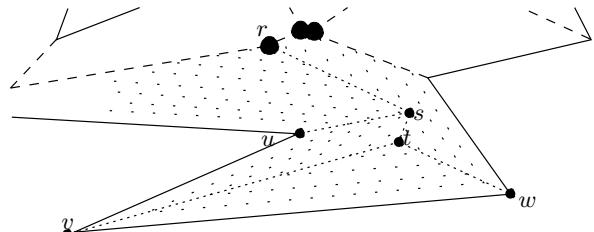


Figure 3: Setting vertex heights according to their offset distance from the original polygons

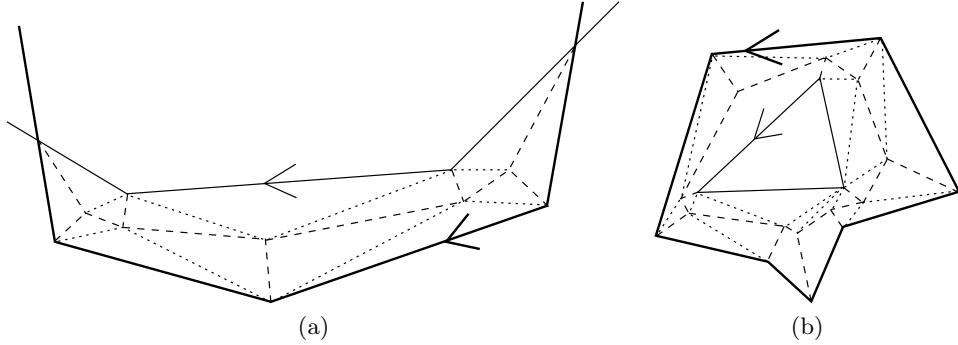


Figure 2: Active cells: Their straight skeletons and triangulations

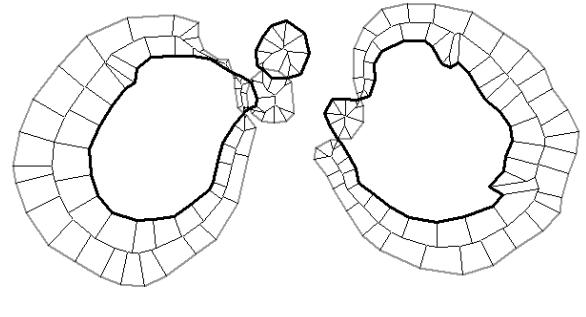
are slid (and are possibly merged or split) along the skeletal edges. The speed of the sliding is proportional to the slope of the respective (lifted up) edges. This is equivalent to intersecting the interpolating surface with a series of parallel planes.

2. SOFTWARE AND VIDEO DETAILS

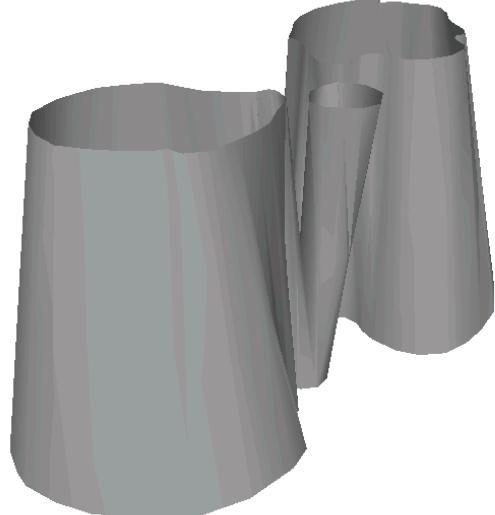
The video segment displays a few sessions of running the software implemented by the second author. The program was implemented in the C++ programming language using Microsoft Visual C++ 7.0 (.NET). It runs on every Win32 environment (95+/ME/XP, NT 4.0). The user interface is based on the MFC library, while the 3D visualization uses OpenGL facilities. The GPC (Generic Polygon Clipper) library (courtesy of Alan Murta) was used for polygon boolean operations. The source consists of about 5,000 lines of code. Figure 4 shows two screen snapshots of our system.

3. REFERENCES

- [1] O. AICHHOLZER, D. ALBERTS, F. AURENHAMMER, AND B. GÄRTNER, A novel type of skeleton for polygons, *J. of Universal Computer Science*, 1 (1995), 752–761.
- [2] G. BAREQUET, M.T. GOODRICH, A. LEVI-STEINER, AND D. STEINER, Straight-skeleton based contour interpolation, *Proc. 14th Ann. ACM-SIAM Symp. on Discrete Algorithms*, Baltimore, MD, 119–127, 2003.
- [3] P. FELKEL AND Š. OBDRŽÁLEK, Straight skeleton computation, *Spring Conf. on Computer Graphics* (L. Szirmay-Kalos, ed.), Budmerice, Slovakia, 210–218, 1998.
- [4] P. FELKEL AND Š. OBDRŽÁLEK, Improvement of Oliva's algorithm for surface reconstruction from contours, *Spring Conf. on Computer Graphics*, (J. Zara, ed.), Budmerice, Slovakia, 254–263, 1999.
- [5] J.-M. OLIVA, M. PERRIN, AND S. COQUILLART, 3D reconstruction of complex polyhedral shapes from contours using a simplified generalized Voronoi diagram, *Computer Graphics Forum*, 15 (1996), C397–408.



(a) Straight skeletons



(b) A reconstructed 3D model

Figure 4: Screen snapshots