Towards an Implementation of the 3D Visibility Skeleton
Linqiao Zhang, Hazel Everett, Sylvain Lazard, Sue Whitesides

To cite this version:


HAL Id: inria-00338095
https://inria.hal.science/inria-00338095
Submitted on 10 Nov 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Towards an Implementation of the 3D Visibility Skeleton

Linqiao Zhang  
McGill University  
Montréal, Canada  
lzhang15@cs.mcgill.ca

Hazel Everett  
LORIA - Université Nancy 2  
Nancy, France  
hazel.everett@loria.fr

Sylvain Lazard  
LORIA - INRIA Lorraine  
Nancy, France  
sylvain.lazard@loria.fr

Sue Whitesides  
McGill University  
Montréal, Canada  
sue@cs.mcgill.ca

ABSTRACT
In this note we describe the contents of a video illustrating an algorithm for computing the 3D visibility skeleton of a set of disjoint convex polytopes. The video can be found at http://www.cs.mcgill.ca/~lzhang15/video/ with file name socg07visidemo.mov.

Categories and Subject Descriptors: F.2.2 [Nonnumerical Algorithms and Problems]: Computations on discrete structures

General Terms: Algorithms

Keywords: Visibility skeleton

1. INTRODUCTION
The 3D visibility skeleton is a graph whose vertices correspond to the maximal free line segments that are transversal to four edges of at least three distinct polytopes and tangent to those polytopes; its arcs correspond to sets of maximal free line segments that are tangent to three polytopes [9]. The visibility skeleton has been used for visibility computations such as computing shadow boundaries [10, 8]. This video demonstrates a sweep plane algorithm for capturing the vertices of the 3D visibility skeleton of a set of polytopes in 3D [13, 5].

2. THE ALGORITHM
The input of the algorithm is a set of k disjoint convex polytopes in general position with n edges in total. The output of the algorithm is the set of O(n^2k^2) vertices of the 3D visibility skeleton of the input polyhedra. The algorithm, which runs in O(n^2k^2 log n) time, can also be used to compute the arcs of the skeleton.

The algorithm performs a rotational plane sweep around each edge e of each polytope, sweeping from one incident face of that edge to the other incident face. The sweep plane intersects the polytopes in at most k disjoint convex polygons, which change their shape as the sweep plane rotates. Figure 1 shows one position of the sweep plane.

Polytope C lies above the plane, with edge e in the plane. Polytope D lies below the plane. Figure 2 shows the view inside the sweep plane. The polytopes A and B intersect the plane in convex polygons A and B, which support 4 bitangents. Figure 3 shows the 2D visibility skeleton corresponding to Figure 2. The circular cycle of directed arcs gives the ordering of the 4 bitangents around polygon A; the cycle of the remaining directed arcs gives the ordering of the 4 bitangents around polygon B.

During the sweep, the algorithm maintains the 2D visibility skeleton of the intersected polytopes [14]. The 2D visibility skeleton for the convex polygons in the initial sweep plane is computed and then used to determine the initial queue of critical events that will occur during the sweep. The sweep planes at which these critical events occur are called event planes. At a critical event, the 2D visibility skeleton may change its topology and the algorithm updates it, as well as the queue of critical events.

There are three types of critical events. A V-event occurs when the sweep plane encounters a polytope vertex that supports one or more bitangents in the sweep plane. A T-event occurs when two or three bitangents become colinear. An F-event occurs when a bitangent becomes colinear with a face of a polytope. There are O(nk^2) events per sweep, and they can be computed and processed in O(nk^2 log n) time in the usual sweep algorithm paradigm.

The vertices of the 3D visibility skeleton are captured during the sweep, as they correspond to the V, T, F-events whose associated bitangents intersect the edge e that the sweep plane is rotating about. After the n sweeps, a description of the arcs of the 3D visibility skeleton can be computed, although the details are not illustrated in the video.

Copyright is held by the author/owner(s).
SCG’07, June 6–8, 2007, Gyeongju, South Korea.
ACM 978-1-59593-705-6/07/0006.
3. IMPLEMENTATION ISSUES AND TECHNICAL DETAILS

Although the algorithm as described in [5] works for any set of possibly intersecting convex polytopes in any configuration, the current implementation requires that the polytopes satisfy certain general position assumptions. The key predicate of the sweep algorithm compares two event planes, to order their occurrences in the sweep. A detailed study of this predicate and its degree is given in [11].

The algorithm was implemented in C++ using the CGAL [6] library. We used the CORE library [7] to perform exact comparisons of algebraic numbers. To compute the 2D visibility skeleton we used the CGAL-based package due to Angelier and Pocchiola [3], based on the Greedy Flip Algorithm [4, 14].

The graphical output was produced using the Geomview software [12] through the interface provided by the CGAL library. We took snapshots of the Geomview window display, while rotating the viewpoint to provide a 3D view of the objects in the display window. Finally, we used iMovie [2] to assemble all the snapshots together into the final video. We used the Audacity [1] software for the audio.

4. ACKNOWLEDGMENTS

We would like to thank Ethan Kim, Christophe Weibel, and Nathan Yu for their help.

5. REFERENCES

[1] Free software distributed under the GNU General Public License (GPL).

1The precise definition of our general position assumptions is straightforward but lengthy. It guarantees, for example, that each critical event corresponds to a unique position of the sweep plane.