3D Visibility made visibly simple: an introduction to the Visibility Skeleton

Frédo Durand, George Drettakis, Claude Puech
iMAGIS/GRAVIR-IMAG*

Many problems in computer graphics require accurate global visibility information, such as the view from a given point, the mutual visibility of two objects (e.g. lighting simulation), the limits of umbra and penumbra, etc. Previous approaches have used coarse approximations or have been complicated to implement, numerically unstable and very expensive both in storage and time. We present a new data-structure, the *Visibility Skeleton*, a multi-purpose tool which can answer many types of queries. It is easy to implement and has been tested with scenes of a few hundred polygons. We present the visual events which encode all visibility changes for polygonal scenes and sketch a construction algorithm as well as queries such as the computation of the part of a polygon visible from a vertex of the scene and the limits of umbra and penumbra between two polygons.

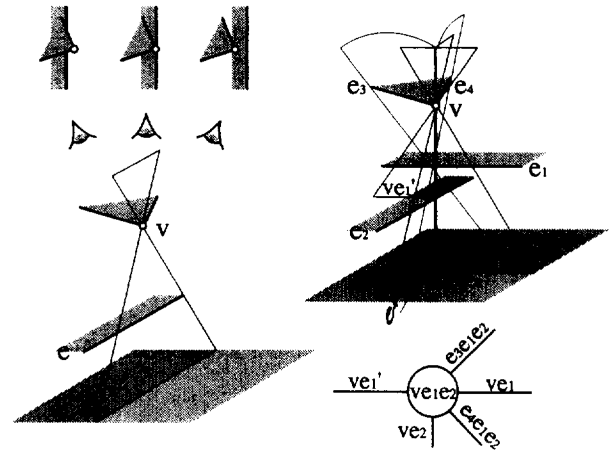
1 Introduction and Previous Work

The computation of accurate visibility information has been addressed in computer graphics, computational geometry and computer vision. In computational geometry, researchers have described algorithms using line sets and Plücker coordinates (see e.g. [Pel90]).

In computer graphics, Teller [Tel92] has used these concepts to compute the limits of penumbra cast by an area light source and Drettakis and Fiume [DF94] and Stewart and Ghali [SG94] have computed *discontinuity meshes* which are the limits of umbra and penumbra together with *back-projections* which encode the topological aspect of the light source for a region of the scene where it does not change.

In computer vision researchers have long been interested in computing the aspect graph which is a partition of the viewing space according to the topological aspect of a view (see e.g. [GM90, PD90]). To compute such a partition, Plantinga and Dyer [PD90] define a data-structure called the *asp* which is a subdivision of the 4 dimensional space of lines for orthographic projection, and a subdivision of the 5 dimensional space of rays for the perspective projection.

*BP 53, 38041 Grenoble Cedex 09, France
Frederic.Durand@imag.fr, http://w3imagis.imag.fr
iMAGIS is a joint project between CNRS/INRIA/INPG/UJF



(a) (b)

Figure 1: (a) EV critical line swath. (b) VEE extremal stabbing line and its adjacent critical line swaths; below, the graph induced.

In 2D, Pocchiola and Vegter have defined a data-structure called the *visibility complex* (see e.g. [PV96, DP95]) which encodes all the visibility information of a scene. This data structure has been used for lighting simulations in flatland [DORP96], and a 3d generalisation has been proposed but not implemented [DDP96].

In this video we introduce a simplification of the 3d Visibility complex which is easy to implement since we only deal with the 0 and 1-dimensional faces.

2 The Visibility Skeleton

Visibility changes can be characterized by *critical line sets* or *line swaths* and by *extremal stabbing lines*.

When the eyepoint traverses a line swath the topology of the view changes. For example, in fig 1(a), the vertex v is initially projected onto the floor, and subsequently projected onto the polygon adjacent to the edge e.

A line swath (or critical line set) is a 1 dimensional set of lines defined by its generators, (v and e in fig 1(a)). In addition there are also EEE swaths which are ruled quadrics.

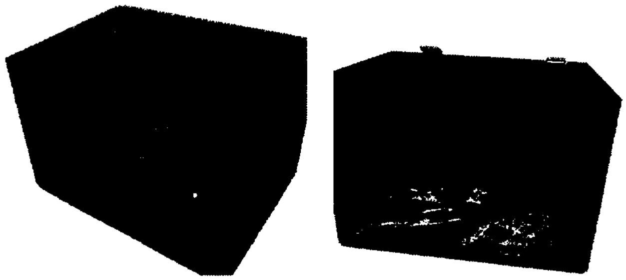


Figure 2: (a) Part of the left wall seen from the vertex on the chair (b) Limits of umbra and penumbra (*discontinuity mesh*) cast by the upper right lamp onto the floor.

Critical line sets meet at *extremal stabbing lines*. In fig 1(b), a VEE extremal stabbing line is shown together with its adjacent line swaths (three EV and two EEE). We define a graph structure (shown in the lower part of fig 1(b)) whose nodes are the extremal stabbing lines and whose arcs are the critical line swaths.

All the lines of a line swath have the same objects (possibly at infinity) at their extremities. We store the arcs of our graph in a two-dimensional array indexed by their extremities to perform efficient queries.

We call this data-structure the *Visibility Skeleton* [DDP97]. It corresponds to the 0 and 1-dimensional faces of the 3D-Visibility Complex [DDP96]; this explains its name.

3 Algorithms

3.1 Construction

The potential nodes of the Visibility Skeleton are enumerated using nested loops on their generators. They are subsequently tested for occlusion using a ray-caster which also finds their extremities. Each node is then inserted in the graph. For each adjacent arc, we check if it has already been created. If not, it is created. Otherwise the arc is split if appropriate. The node can then been attached to the arc.

This algorithm has $O(n^5)$ time complexity, but this is a very pessimistic bound, and simple culling techniques allow the practical running time to grow about quadratically in the few examples we have tested. See [DDP97] for more details.

3.2 Queries

To compute the part of a polygon p seen by a vertex v, we consider the arcs for which p is an extremity and v is a generator. An example is shown fig 2(a).

The limits of umbra and penumbra cast by a polygon considered as a light source upon another polygon of the scene correspond to the arcs whose extremities are those two polygons, or the arcs which have a generator on the light source and an extremity on the other polygon. An example is shown fig 2(b).

4 The video

The video introduces the notions of critical line sets and extremal stabbing lines. The construction algorithm is sketched and some queries are demonstrated. The initial scene has been chosen since, despite being simple, it includes a wide range of types of critical line swaths and extremal stabbing lines, while the second scene (fig 2) is a typical computer-graphics scene of 450 polygons.

The construction of the skeleton and the queries were implemented in C++ on a SGI workstation using the library Open Inventor for visualization.

The video was recorded in real time and transfered on a Macintosh where the editing was performed with Adobe Premiere and Photoshop. Graphs and arrays were added manually in Premiere, as were animations such as the arrows for ray casting.

Author/Title Index

[DDP96] Frédo Durand, George Drettakis, and Claude Puech. The 3d visibility complex, a new approach to the problems of accura te visibility. In X. Pueyo and P. Shroeder, editors, *Proceedings of 7th Eurographics Workshop on Rendering in Porto, Portugal (Rendering Techniques '96, Springer Verlag)*, pages 245–257. Eurographics, June 1996.

[DDP97] Frédo Durand, George Drettakis, and Claude Puech. The visibility skeleton: A powerful and efficient multi-purpose global visibility tool. In *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '97 (Los Angeles). ACM, August 1997.

[DF94] George Drettakis and Eugene Fiume. A fast shadow algorithm for area light sources using back projection. In *Computer Graphics Proceedings, Annual Conference Series:* SIGGRAPH '94 (Orlando, FL), pages 223–230. ACM SIGGRAPH, New York, July 1994.

[DORP96] Frédo Durand, Rachel Orti, Stéphane Rivière, and Claude Puech. Radiosity in flatland made visibly simple. In *video 12th Annu. ACM Sympos. Comput. Geom.*, 1996.

[DP95] Frédo Durand and Claude Puech. The visibility complex made visibly simple. In *video 11th Annu. ACM Sympos. Comput. Geom.*, 1995.

[GM90] Ziv Gigus and Jitendra Malik. Computing the aspect graph for the line drawings of polyhedral objects. *IEEE Trans. on Pat. Matching & Mach. Intelligence*, 12(2), February 1990.

[PD90] H. Plantinga and C. R. Dyer. Visibility, occlusion, and the aspect graph. *Internat. J. Comput. Vision*, 5(2):137–160, 1990.

[Pel90] M. Pellegrini. Stabbing and ray shooting in 3-dimensional space. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 177–186, 1990.

[PV96] M. Pocchiola and G. Vegter. The visibility complex. *Internat. J. Comput. Geom. Appl.*, 96. special issue devoted to ACM-SoCG'93.

[SG94] A. James Stewart and Sherif Ghali. Fast computation of shadow boundaries using spatial coherence and backprojections. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 231–238. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[Tel92] Seth J. Teller. Computing the antipenumbra of an area light source. *Computer Graphics*, 26(4):139–148, July 1992. Proceedings of SIGGRAPH '92 in Chicago (USA).