Accurate Scene Display by Using Visibility Maps

Jérome Grasset, Olivier Terraz, Jean-Marc Hasenfratz, Dimitri Plemenos

Accurate Scene Display by Using Visibility Maps

J.Grasset[1], O. Terraz[1], J.-M. Hasenfratz[2], D. Plemenos[1]
[1]*MSI Laboratory, University of Limoges*
[2]*IMAG Laboratory, University of Grenoble*

Abstract

A tool permitting to improve various steps of the visualisation process is presented in this paper. More precisely, this tool should concern the phase of preparing visualisation and also the visualisation phase itself. The tool presented in the paper, named visibility map, is based on the use of planar maps containing information on visibility and lighting, together with adjacency information inherent to planar maps. The proposed structure should permit to simplify the user's work, to accelerate some phases of visualisation and to improve the quality of produced images. In particular, this method permits to display a scene with ray tracing-like quality images, with very efficient antialiasing.

Keywords: Topological modelling, Planar maps, Visibility maps, Rendering, Antialiasing.

1. Introduction

The work we propose consists of computing a visibility map for a scene, i.e. a list of visible polygons or portions of polygons, structured like a planar map. An application proposed here is to obtain good quality image, taking into account shadows, reflection, penumbra, with an efficient antialiasing. This can be used, for example, to fast preview a scene in CAD. Other applications of visibility maps are envisaged, as explained in section 6.

There exists a lot of algorithms for resolving the hidden surface removal problem ([Dorw 94] is a survey of object-space ones). The closest to our method are algorithms using a depth sort of the scene's polygons, in order to display polygons from the farest to the closest one [NNS 72] and to obtain an image where the non visible parts of polygons are hidden. Based on the same approach, priority trees [FKN 79] give a structure to the scene and permit real time display, even when the viewpoint is modified. Other methods, as beam tracing [HH 84], try to compute precisely visible polygons. However, the set of obtained polygons is not really structured.

Recent works in the domain of discontinuity meshing use structured sets of polygons for clipping the scene according to discontinuity lines of the radiosity function, especially shadow boundaries, like [LTG 92], [DF 94], [DDP 97]. The difference with the method proposed here is that its purpose is not the same, so discontinuity lines are not explicitly used to guide the clipping.

Another technique for computing visibility has recently been proposed [SK 98]. This technique computes an approximate visibility map by reconstructing polygons from a more or less precise projection of the scene's polygons.

The method we will develop in the following sections tries to combine exact visible polygon computing and strong structuring of the set of visible polygons.

After a brief definition of topological modelling and planar maps in section 2, we will introduce, in section 3, the concept of visibility maps, permitting to take into account shadowing, transparency and reflection. The use of visibility maps to obtain high quality images of a scene is detailed in section 4. In section 5 advantages and drawbacks of the proposed method are discussed and some results are presented while in section 6 we discuss other possible applications of visibility maps. Finally, section 7 permits to conclude.

2. Topological modelling and operations on maps

2.1 Topological Modelling

Topological modelling permits to dissociate, during scene modelling, topological information (adjacency and incidence relations between cells, number of holes, ...) from purely geometric information like coordinates of vertices. We used a two dimensions topological model.

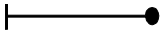


Figure 1: a dart.

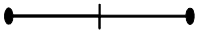


Figure 2: an edge.

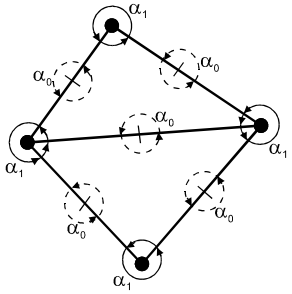


Figure 3: a map.

[1] {grasset | terraz | plemenos}@unilim.fr
[2] Jean-Marc.Hasenfratz@imag.fr

Thus, cells are vertices, edges and faces that we associate to points, segments and polygons of the image space.

The model used here is very close to the model of combinatorial maps which is equivalent to topological model is the winged-egde data structure proposed in [Baum 72] (see [Lien 90] for a survey of topological models). The primitive elements are darts, which can be viewed as half-edges (see figures 1, 2 and 3), connected by means of α_0 links (to create edges) and α_1 links (to organize edges in polygons). This model does not duplicate edges and is perfectly adapted to refinement operations, described in the following paragraph, with a very simple organisation of topology, based on intersection points.

2.2 A Very Important Operation: Refinement Of Two Maps

A map is considered correct if there is no intersection among its edges. In order to process two overlapping maps and to obtain a new correct map (refinement of two maps), we look first for all intersection points among all edges of the two maps. To do this, we use the scan-line method described in [BvKOS97], but we include the processing of special cases like horizontal or overlapping lines, absolutely necessary for our method. This algorithm is intersection sensitive: the running time is linearly dependent on the number of intersections and linearly dependent on the number of edges. The general principle of the method is to maintain a list of active edges and to use a scan line for each vertex of the map, including new intersection points. Intersection tests are performed each time a starting point of a new edge is detected, but a very restricted number of edges is concerned.

During the intersection points detection process, α_1 links are updated, in order to achieve the process with a correctly organised map. This map does not yet contains all required information for each polygon. Thus, a second step is necessary in order to associate with each modified polygon its new properties (which object or which lighting does the polygon belong to, and so on). This operation is performed by looking for darts of the map, sharing new or modified edges.

It is also necessary to update information of inclusion of a polygon in another one. By using polygon properties, known at this level, the processing is simple and robust. Indeed, we have a set of polygons which do not have real intersection, except possible common edges or vertices. The test permitting to know whether a polygon A is included in a polygon B or not is reduced to verifying whether a point strictly contained in the polygon A is also contained in the polygon B or not. No particular cases are to study. The only problem is the determination of a point contained in a polygon, because of numerical precision limits in the case of small polygons.

2.3 Operations Between Maps

In order to achieve the construction of visibility maps, we need three operations among maps: clipping, masking and adding. These operations can be easily deduced from the refinement.

Clipping (figure 4) permits to conserve only parts of polygons which are really in the image. The projection is clipped according to desired contour.

Masking (figure 5) is based on depth sort of polygons: new polygons are integrated to a given map, starting from furthest one. Each new integrated polygon masks the polygons already integrated.

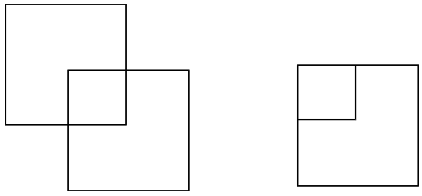


Figure 4: clipping of polygon B by A.

Adding (figure 6 and figure 11) is used to overlap two maps with conservation of all information on each map. This operation is useful when we wish to integrate a shadow map in an already existing map. All the new intersection polygons must be conserved.

These operations are performed by using propagation of marks on the darts during the refinement process,

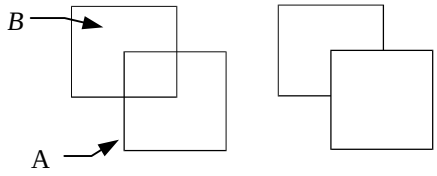


Figure 5: masking B with A.

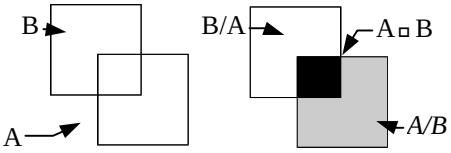


Figure 6: adding A to B.

simply by traversing darts and without need of geometric tests. The used method is inspired from [Cazi 97] and is based on adding of encountered labels during the processing of a polygon. Each dart contains, before the refinement, the mark of the polygon which it is associated to. During the refinement of two maps, a dart can receive an additional mark if its edge overlaps an edge of another map. At the end of the process, the darts containing two marks belong to the intersection of two maps. Marks are then propagated to all the darts of the new polygons in order to obtain a map indicating the provenance of each new polygon. Now, we can perform all desired operations.

3. Visibility maps

The visibility map we are going to present here gives an account of visible polygons from a particular viewpoint and, also, of the shadows. It includes too topological informations meaning adjacency and incidence relations between these elements. It can be viewed as a representation of the image obtained from this viewpoint but without any choice of resolution . The map remains unchanged if a new display is performed with a higher resolution. The current definition of visibility maps concerns only scenes composed of polygons.

The general method of construction of a visibility map is the following:

- Sorting of polygons
- For each polygon to project do:
  - Compute reflections on this polygon (recursive process) and construct a map of reflection
  - Compute transparency and overlap with the map of reflection
  - Compute shadows and overlap with the visibility map
  - Project the polygons and the maps it supports.

3.1 Visible objects, directly or by reflection and transparency

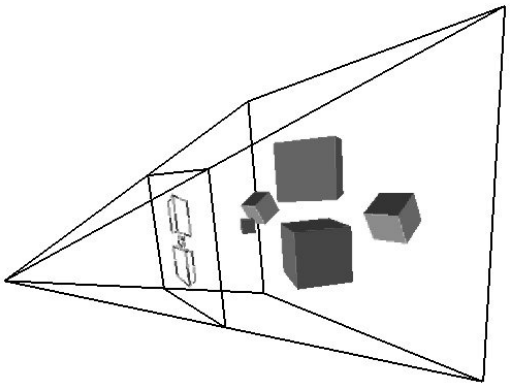


Figure 7: visibility map

The first step of the visibility computation process is the determination of directly visible polygons of the scene. To do this, a depth sort of the scene's polygons is performed first. Then, the polygons of the scene are projected in the image plane, starting from the furthest one and ending with the nearest. Each projected polygon is integrated in the existing map by using the masking operation, after a possible clipping (see figure 7and 12).

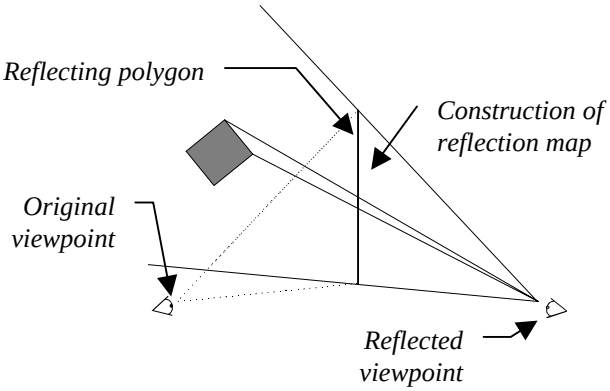


Figure 8: processing of reflections

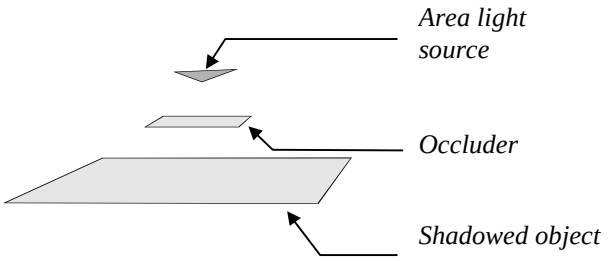
The next phase of the visibility map construction process is the determination of indirect visibility, which takes into account reflections and transparencies. To obtain visible parts of the scene by reflection on a polygon, we consider a new virtual view point which is the image of the real view point obtained by reflection on this polygon (see figure 8). A map is then computed on the reflecting polygon using the visibility map construction method and this map is finally projected in the current image plane. This process is recursive: if a reflecting polygon is viewed by reflection on another polygon, the map of this secondary reflection must be computed.

It is not possible to take into account refraction with our method. Indeed, the image by refraction of a polygon is not a polygon and cannot be used to construct a visibility map. Approximate methods could be used by decomposing the refracting polygon into «small» polygons, in order to consider that the image of the current viewpoint is a point : the aim is to obtain a set of situations where the usual Gauss approximations are convenient (see for example discussion in [HH 84]). However, these methods are not satisfactory because they take into account the notion of dimension and the produced visibility map would be dependent on image resolution.

3.2 Shadows And Penumbrae

The construction of a visibility map must also take into account lighting effects. The final result will be a visibility map whose each polygon represents a unique portion of the three-dimensional scene (viewed directly or indirectly) and receiving a unique type of lighting.

To determine shadows produced by punctual light sources, we construct a new map, so called «shadow map», on each concerned polygon of the scene. The shadow map construction process is similar to the one used to determine visibility: the polygons are sorted according to their distance from the current polygon and projected in this polygon, starting from the furthest one.



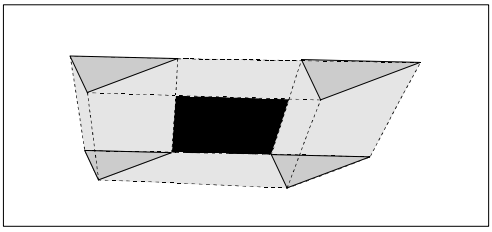


Figure 9: processing of area light sources

For area light sources, a method similar to the method proposed in [NN 83] is used. So, the obtained clipping takes into account shadow and penumbra areas (see figure 9). A detailed description of penumbra boundaries computation can be found in [Tell 92].

4. Displaying an image from a visibility map

4.1 The displaying method

In order to get an image from a map, a discretisation step is chosen, i.e. a resolution for the final image. A grid is created which is a discrete map of the edges of the visibility map. Each active element of the discrete edge grid contains a reference to the previous polygon, a reference to the next polygon and the ratio of its belonging to the previous polygon. The elements to activate are determined using the Bresenham's line tracing method.

The discrete edge map and the pixel grid (image space) are scanned from left to right. For each element of the discrete edge map the current polygon is updated, if the element is an active one, and the colour of the current pixel of the pixel grid is determined from the knowledge of current polygon. For the elements of the pixel grid corresponding to active elements of the discrete edge map, colour is determined by using information on previous polygon, next polygon and belonging ratio. In this manner, efficient anti-aliasing is performed.

The colour of each pixel is determined by ray tracing but the used method is very fast because of knowledge on intersected polygons by a ray and on received lighting. Thus, it is useless to trace rays which do not intersect objects, to do intersection test with non-intersected objects or to verify whether an intersection point is in a polygon or not.

4.2 Multiple renderings

One of the advantages of the method is the possibility to have successive renderings with some parameters (quality of lightings, scene material, ...) modified. As the display is very fast, it is easy to apply operations which do not modify the topology of the map. Thus, it is possible, for example to iteratively determine a good lighting of the scene whose components (objects, light sources, observer) have fixed positions.

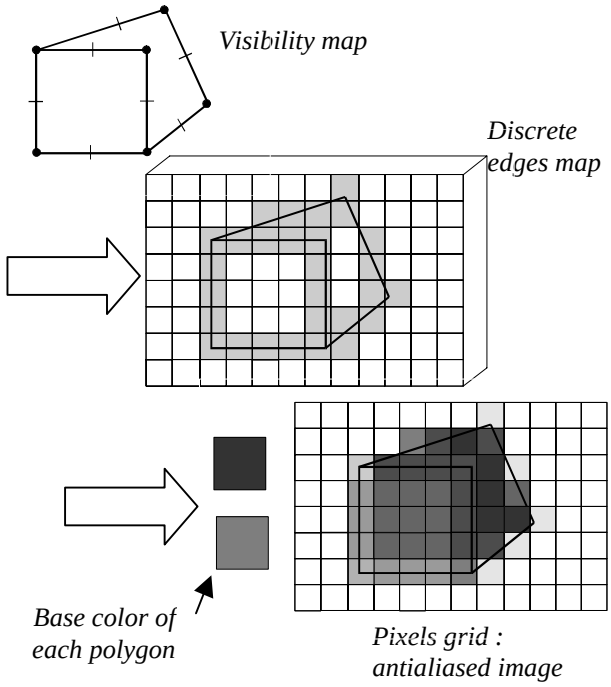


Figure 10: the displaying method

5. Results and discussion

The data stored in the visibility map (the adjacency information in particular) allow an efficient antialiasing (see figure 14) of the edges with very few additional computation time. This provides better and faster results than traditionnal super-sampling methods. Since the rendering of the image is fast, it is possible to change resolution (to zoom in or out) and perform the rendering again which avoid any image deterioration.

We have implemented refinement, operations between maps (see figure 11, 12 and 14), and scan line process for rendering (see figure 14). Shadows and reflection computation have already been tested with another topological model that we gave up because it was not efficient enough for combining maps. Some intermediate techniques, such as beam tracing used to sort polygons before projection ([HG 99]), will be replaced or improved. We are probably going to use BSP structures, presented by [FKN79], or other hierarchical polygons representation (for instance [CW 96]): this would allow to compute map faster when the viewpoint

changes.

The visibility map solves or avoids some numerical approximation problems, with a structure insuring that there are not unclosed polygons, intersecting edges within the same polygon, or other kinds of topological errors. So the topological approach seems to be more interesting for clipping than methods that use only sets of polygons like in [WA 77]. However, because of the mergings performed during the construction of the map, some « small » objects or details may be lost. This might be particularly important if the map is used to perform quick « zoom in »: a close view of the image may be false.

The method we proposed here is viewpoint-dependent (see figure 12 and 13). The shadow and penumbra maps can be computed just once and left inside the scene, on the polygons they are related to. But the direct and reflected visibility must be computed each time the viewer position changes: it would be interesting to study methods allowing to re-use some of the already calculated informations, in particular in case of continuous moves.

The preprocessing costs might be considered as an drawback of our method. In fact, that depends on the use of the map. If the user just wants to render one image, the preprocessing might be considered too heavy (more especially as some possibilities of ray tracing are lost, like refraction simulation) even if it allows some quality improvements. But if the user wants to see several renderings with the same map (for example to adjust light) the relative preprocessing cost will be reduced: the more the map is used, the less expensive the preprocessing is. That is a motivation to develop applications of the visibility and shadow maps.

6. Future works and applications

The method presented is this paper is used to improve scene rendering. We are working on other applications which reduce relative cost of map preprocessing in the whole process.

6.1 Scene design

In order to help the scene designer, it could be possible to ask him to designate polygons of the scene to be illuminated (or shaded) and then compute areas of the scene where light sources might be put. A mean to do this is to compute a shadow map on each polygon, considering areas to be illuminated as light sources: the regions of polygons illuminated by these "fake" light sources are regions which "see" areas to be illuminated.

6.2 Preparing scene for rendering

After modelling, some preparations might be required before rendering: for radiosity computation it is necessary to split polygons into patches. For efficient meshing it is interesting to follow the discontinuity lines of the radiosity function. This insures better homogeneity of the energy on each mesh. With the shadow and penumbra maps we construct, we can perform discontinuity meshing: the meshes just have to follow the boundaries of the faces of the map.

6.3 Scene adjusting and image understanding

When the visibility map is calculated, some changes of the scene do not modify it. It is possible, for example, to change light intensity or objects colors: this does not change the topology of what we see, so the visibility map is the same. We can exploit this fact by letting the user adjust this parameters and giving him quick and accurate results.

Another application is to help the user to understand a scene. We know exactly where each face of the map comes from, so we can propose these informations to the image viewer in order to help him to understand what he sees. We can explain which object of the scene a face is projected from, if this object is seen directly, by reflection or transparency or which light sources illuminate it.

Since the resolution of the image is chosen after the map has been constructed, it is obvious that we can change the resolution without changing the map. By the way we can make quick and fairly accurate zoom of any part of the image. This zoom cannot be considered as perfectly accurate because of the precision limit of the computation of the map: some small polygons features or some small objects may be lost, and when we increase the size of the picture they are still missing even if they should occupy a large area of the image.

7. Conclusion

We proposed a method to construct accurate visibility map, including shadows, reflections and transparencies. With this only tool we can develop several applications concerning user's help, discontinuity meshing, light sources adjusting, fast and antialiased ray-tracing. The topological approach reduces problems of numerical approximations, ensuring correctness of the calculated faces.

The visualisation process proposed here gives easy and efficient antialiasing and allows interesting image resizing. Nevertheless, compared to ray tracing usual features, we cannot simulate refraction and we just take into account objects made of polygons.

The aim of future works will be to develop other applications to take benefits of informations included in the map, concerning adjacency or knowledge about the 3D scene.

Figures

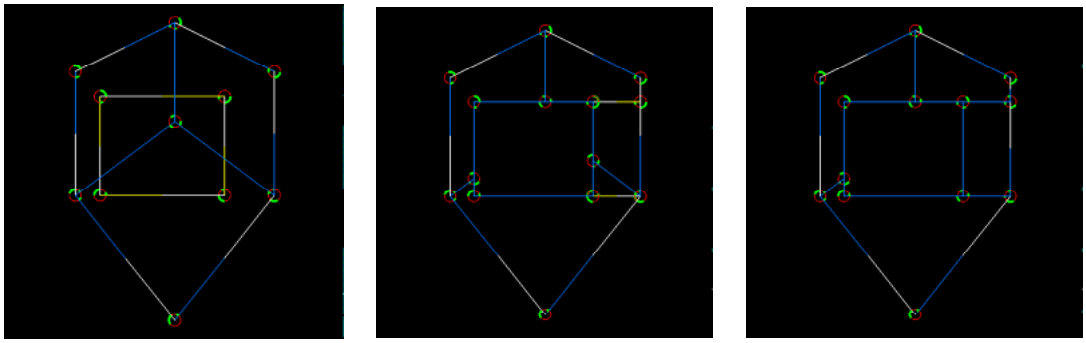


Figure 11: an example of refinement that shows two maskings of the existing map by projected polygons

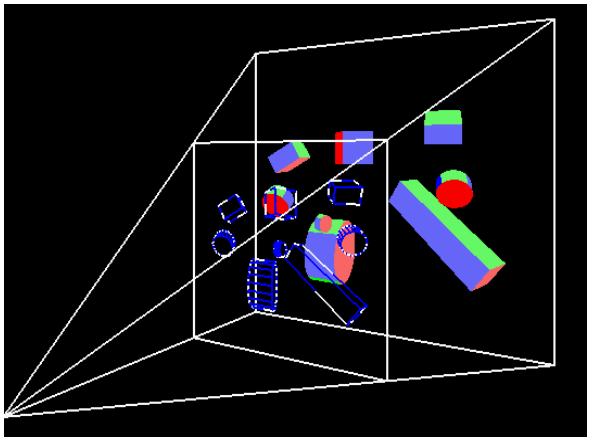


Figure 12: computing a visibility map

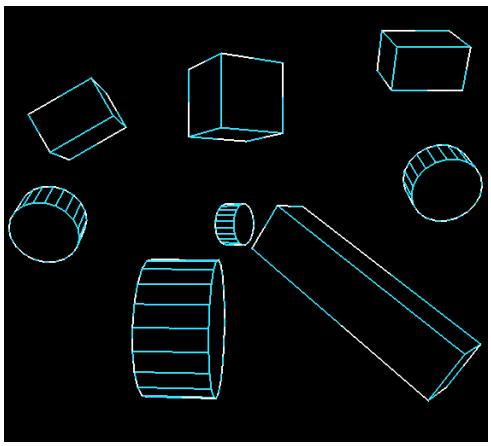


Figure 13: the visibility map of the previous scene

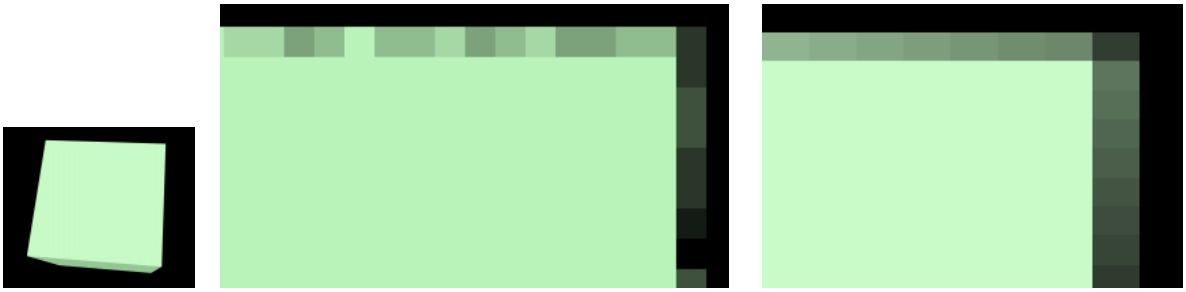


Figure 14: antialiasing of a cube (left), obtained with the ray tracing software Pov Ray (centre) and with a visibility map (right). The antialias threshold value used with POV is the value recommended in its documentation.

References

[Baum 72] B.G. Baumgart
Winged-edge Polyhedron Representation. Technical Report STAN-CS-320, Standford University, 1972

[BvKOS 97] M.de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf.
Computational Geometry, Algorithms and Applications Chapter 2: « Line segment intersection, thematic map overlay ». Springer-Verlag, Heidelberg, pp.19-42, 1997.

[Cazi 97] D.Cazier
Construction de systèmes de réécriture pour les opérations booléennes en modélisation géométrique. Thèse de doctorat, University of Strasbourg, 1997

[CW 96] H.-M. Chen, W.-T. Wang
The feudal priority algorithm on hidden surface removal. SIGGRAPH proceedings, 1996

[DDP 97] F.Durand, G.Drettakis, C.Puech
The Visibility Skeleton: a powerful and efficient multi-purpose global visibility tool.
Computer Graphics proceedings, 1997

[DF 94] G.Drettakis, E.Fiume
A fast shadow algorithm for area light sources using backprojection Computer Graphics proceedings, 1994

[DKW 85] N.Dadoun, D.Kirkpatrick, J.Walsh
The geometry of beam tracing ACM Symposium on Computational Geometry, Vol 23, pp. 335-344, 1985

[Dorw 94] Susan E. Dorward
A survey of object-space hidden surface removal. International Journal of Computational Geometry and Applications, Vol.4, No. 3, pp. 325-362, 1994

[FKN 79] H.Fuchs, Z.M.Kedem, B.F.Naylor
On visible surface generation by a priori tree structures. ACM, 1979

[HG 99] J-M.Hasenfratz, D.Ghazanfarpour
Une synthèse des variantes du lancer de rayons et du lancer de faisceaux
To appear in Revue internationale de CFAO et d'Informatique Graphique, 1999

[HH 84] P.Heckbert, P.Hanrahan
Beam tracing polygonal objects Computer Graphics, 18(3), pp. 119-127, July 1984.

[Lien 90] P.Lienhardt
Topological models for boundary representation: a comparison with n-dimensional generalized maps Computer-Aided Design, vol 23, n°1,, pp. 59-82, 1991

[LTG 92] D.Lichinski, F.Tampieri, D.P.Greenberg
Discontinuity meshing for accurate radiosity
IEEE, November 1992

[NN 83] T.Nishita, E.Nakamae
Half tone representation of 3-D objects illuminated by area sources or polyhedron sources. COMPSAC'83, Proc. IEEE 7th Intl. Conf. Soft. and Appl. Conf., pp. 237-242, 1983

[NNS 72] Newell M.E., Newell R.G., Sancha T.L.
A new approach to the shaded picture problem Proceedings. ACM National Conference, 1972

[SK 98] A.J.Stewart, T.Karkanis
Computing the approximate visibility map, with applications to form factors and discontinuity meshing. Eurographics workshop on rendering, 1998

[Tell 92] S.J.Teller
Computing the antipenumbra of an area light source. SIGGRAPH proceedings, 1992

[WA 77] Weiler K., Atherton P.
Hidden surface removal using polygon area sorting. SIGGRAPH ACM, Computer Graphics, vol. 11, n°2, 1977