On Mesh Preprocessing for Radiosity

Jan Přikryl

Czech Technical University, Prague
e-mail: prikryl@sgi.felk.cvut.cz

Abstract: In this poster we discuss methods of radiosity mesh preprocessing for more accurate radiosity solutions. We present a mesh preprocessing approach improving the geometry description of the scene data and exploiting a restricted set of discontinuities or the geometrical approach for shadow leakage removal.

Keywords: Radiosity, mesh, discontinuity meshing.

1 Introduction

Common radiosity programs require a mesh of planar elements representing all scene surfaces as their input. Since the mesh is a discrete description of the scene only and we have to interpolate the computed radiosity values between the mesh points or elements, we may miss some details in the radiosity distribution and the resulting image may contain some errors caused by unproper meshing.

The aim of this poster is to present some ways how to improve the mesh during the preprocessing phase in a reasonable time.

2 Implementation

Our preprocessing has three phases: In the first one we preprocess the scene data and build a BSP tree over it. In the other phases we use this tree to provide either restricted discontinuity meshing or the shadow leakage removal.

2.1 Initial data preprocessing

The process of the input mesh removal is composed of six steps: In the first step we try to merge all multiple defined vertices and represent them as one vertex only. During this step we also take care of non-aligned vertices which can appear in input data – we use a small fixed tolerance in vertex position when looking it up in a list of already accepted vertices.

In the second step we merge the original faces into maximal planar patches with holes containing possible spikes, loops, and divided edges (see Figure 1). Two patches are merged together if they are in contact and their normal vectors are "approximately" equal. A similar approach has been already presented in [6].

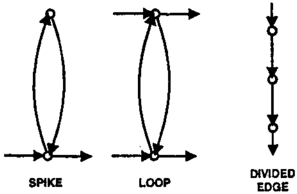

Figure 1: Mesh artifacts

In the third and fourth steps spikes and edge loops are removed, and divided edge parts are joined together skipping the splitting vertices.

In the fifth step, free vertices are removed from the scene data.

In the last step we break all the polygons into a set of convex patches suitable for building the BSP tree. Our algorithm of convex polygon generation starts at an arbitrary concave vertex of a tested polygon. Then we visit next and previous vertices of the polygon counting the number of vertex "hops" until we visit a concave vertex or until the edge connecting the original and tested vertex creates a concave polygon again. We obtain the number of forward and backward "hops" and the candidate which has the greatest "hop count" (i.e. that one which creates a polygon containing more edges) is then chosen.

2.2 Building the BSP tree

The process of building a BSP tree for a set of polygons is well known [1]. Taking in account the finite numerical precision of computations and the possibility of plane alignment errors we have to use tolerances to compare which planes should form the actual BSP tree node and which are outside this node.

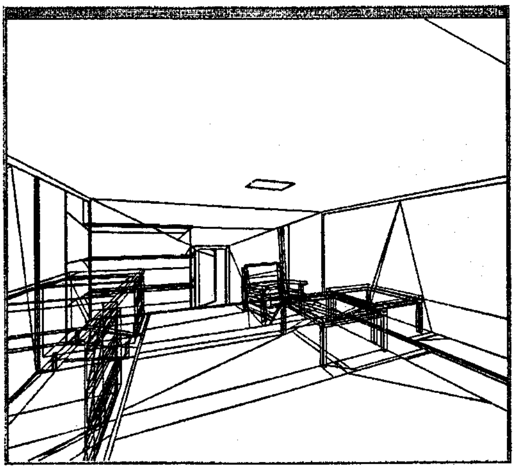

Figure 2: Restricted discontinuity mesh

2.3 Restricted set of discontinuity edges

The mesh preprocessing with respect to the light sources provides us with an accurate yet rather complex mesh. Lischinski et al. [5] solved this problem by generating discontinuities for the actual radiating patch only. In the preprocessing phase this is not possible since we have to process the data as a whole — we therefore tried to reduce the number of patches by choosing only primary light sources and by skipping some perceptually less important discontinuities.

We use a simple thresholding scheme: as a wedge is computed we may obtain the length of the discontinuity line segment, the ratio of distances between the light source vertex and the discontinuity, and between the light source and the occluding edge. If the discontinuity line segment is shorter than some epsilon and the distance ratio is not close to one we do not insert this discontinuity line into the scene mesh (see Figure 2).

This way the size of generated data set decreases, but we skip details which should not be so important.

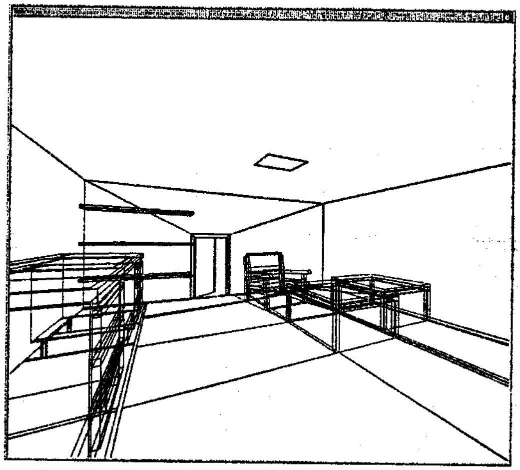

Figure 3: Mesh removing the shadow leakage effects

2.4 Geometrical preprocessing

The main disadvantage of a discontinuity preprocessing algorithm is its time complexity. If we can accept a worse quality of the final image a faster solution would be not to create the discontinuity mesh and only to remove places where the shadow leakage occurs.

If the scene is correctly specified from the geometrical point of view then the BSP tree built for discontinuity preprocessing can be used also here. The shadow leakage may occur only at places where two surfaces are "in contact", close enough to be stored in a single BSP node. If this happens then one or both of these patches are cut along lines representing the borders of patches (see Figure 3).

3 Results

The above mentioned ideas were implemented as a UNIX filter in ANSI C and tested on SGI workstations running IRIX 5.3. The time measurements presented in Table 1 were obtained through UNIX `clock()` facility and show total system times needed for the completion of particular parts of the tested program on an Indigo2 workstation equipped with R4400 processor running at 200MHz. Since UNIX is a multiprocess and multiuser operating system the real execution time is of course higher.

Scene file	Number of patches	Scene reading (s)	Mesh preprocessing (s)	Discontinuity meshing (s)	Shadow leakage (s)
test.obj	24	0.040	0.010	0.490	0.060
buero.obj	510	1.240	0.280	39.860	2.880

Table 1: Time measurements for two sample scenes

References

[1] 3D FAQ. `http://www.csh.rit.edu/~pat/misc/3dFaq.html`, 1996.

[2] D. R. Baum, S. Mann, K. P. Smith, and J. M. Winget. Making radiosity usable: Automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. In *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, pages 51–60, July 1991. Published as Computer Graphics (ACM SIGGRAPH '91 Proceedings), volume 25, number 4.

[3] A. T. Campbell and D. S. Fussell. Adaptive mesh generation for global diffuse illumination. In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, pages 155–164, Aug. 1990. Published as Computer Graphics (ACM SIGGRAPH '90 Proceedings), volume 24, number 4.

[4] N. Chin and S. Feiner. Fast object-precision shadow generation for areal light sources using BSP trees. In D. Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, pages 21–30, Mar. 1992.

[5] D. Lischinski, F. Tampieri, and D. P. Greenberg. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications*, 12(6):25–39, Nov. 1992.

[6] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In E. E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 65–70, July 1992.

[7] M. Segal. Using tolerances to guarantee valid polyhedral modeling results. In F. Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 105–114, Aug. 1990.