The Close Objects Buffer: A Sharp Shadow Detection Technique for Radiosity Methods

A.C. Telea, C.W.A.M. van Overveld
Department of Mathematics and Computing Science
Eindhoven University of Technology
P.O. BOX 513, 5600 MB Eindhoven, The Netherlands

Abstract

Detecting sharp illumination variations such as shadow boundaries is an important problem for radiosity methods. Such illumination variations are captured using a nonuniform mesh that refines the areas exhibiting high illumination gradients. Nonuniform meshing techniques like discontinuity meshing often rely on shadow casting, so their application is computationally expensive. This paper presents a sharp shadow detection technique for radiosity tools based on the progressive refinement method. The presented technique offers good results (especially for capturing sharp shadows cast by small 'detail' objects), is very simple to implement, has negligible time and space requirements and integrates well with other adaptive subdivision strategies in a radiosity tool based on progressive refinement.

1 Introduction

Radiosity techniques estimate the illumination in all the points of a tridimensional environment using a global illumination model [4], [8]. Typically the environment is discretized, the illumination is evaluated in the discretization points and the results are interpolated (e.g. by Gouraud shading) to render the environment.

The quality of the rendered images strongly depends on the discretization of the environment into elements. Smaller elements should be used in areas of rapid illumination variation in order to accurately capture this variation.

Producing a nonuniform mesh, however, is difficult since the illumination is not known beforehand but is to be computed using the mesh itself. One of the best-known techniques for producing a nonuniform mesh is discontinuity meshing based on shadow casting. Shadows are cast from the light emitters and the elements intersected by shadow boundaries are subdivided until the radiance variation over their surfaces is considered acceptably small [7], [1].

Because shadow algorithms are expensive, discontinuity meshing based on shadow casting is a slow technique. This is especially true for scenes containing many complex objects and when shadow casting from secondary light sources is used during a progressive refinement radiosity process [2]. Casting shadows only from the initial light sources is a faster but less accurate alternative since all shadows created by highly reflective objects acting as secondary light sources are discarded. Additionally, keeping track of complex shadow boundaries intersecting elements can be a very elaborate process.

Sharp shadows can be detected in many cases without the need of an expensive shadow casting process. There is a high probability to have visible illumination detail like sharp shadows when an element of the environment's mesh is partially occluded from a powerful light emitter by an object which is very close to the element. An example is a desk having a number of small objects laid on it, as paper sheets, illuminated by a desk lamp (Figure 5). If the lamp casts a strong illumination, complex detail sharp shadows will appear on the desk due to the objects laid on it. The shadow's sharpness increases as these occluding objects get closer to the elements of the desk's mesh.

This paper presents the close objects buffer (COB), a simple software tool for detecting sharp shadows caused by occluding objects that are close to the occluded objects. The presented algorithm can in most cases successfully replace shadow casting based discontinuity meshing, has smaller time and space constraints and is considerably simpler to implement in a progressive refinement radiosity method.

2 Description of the Close Objects Buffer

The COB is an item buffer storing references to all objects in the scene that are above and close to the visible surface of an element of the scene's meshing. There is a COB for each element of the environment's subdivision. The function of the COB of an element is to detect the cases when there's a high probability to have a sharp shadow cast by an occluder which is close to the surface of that element. Such a probability exists when the COB contains at least one occluding object (Figure 1).

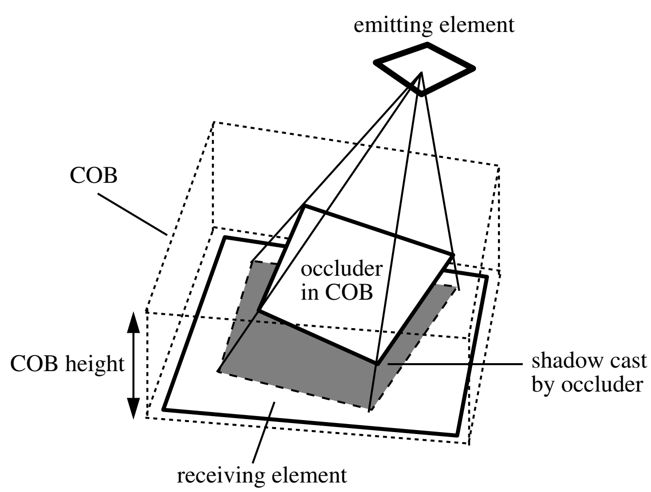


Figure 1: An element and its COB

Geometrically speaking, the COB is a prism similar to the hemicube placed on the top of an element, with the base coincident with the element and the height computed as a fraction of the buffer's element size. The expression of the COB's height relies on the fact that a shadow's sharpness is proportional to the ratio between the emitter-receiver and the occluder-receiver distances. An object is said to be in the COB if it intersects this prism.

The COB of an element can be in three states: uninitialized (no buffer-object intersections have been performed yet), empty (the buffer contains no objects) and not empty (the COB contains at least an object).

The COB algorithm is outlined in Figure 2. At the start of the progressive refinement process, all elements have an uninitialized buffer. When a light emitting element j is about to shoot its radiosity to a receiver i, we first check if the receiver is smaller than a given subdivision threshold, in which case we directly shoot to it. If the receiver is larger than the subdivision threshold, the emitter-receiver interaction magnitude M is estimated using the receiver's total reflectivity $\rho_{R_i} + \rho_{G_i} + \rho_{B_i}$ and the emitter-receiver form factor:

$$M = (\rho_{R_i} + \rho_{G_i} + \rho_{B_i}) \frac{\cos\theta_i \cos\theta_j}{\pi r^2} A_j \tag{1}$$

where θ_i , θ_j are the angles the normals of elements i, respectively j make with the line connecting their centroids, r is the distance between the centroids and A_j is the emitter's area. If M exceeds a given threshold, the receiver is strongly illuminated by the emitter, therefore possible occluders can cast visible shadows. The presence of such occluders is found using the receiver's buffer COB_j. If the COB is uninitialized, this is done now. If the COB is empty, there aren't any occluding objects that might cast sharp shadows over the element, so the emitted radiosity is received by the element. If the COB is not empty, there are objects partially occluding the strong light coming from the emitter and potentially casting sharp shadows. The receiver is subdivided without shooting at it anymore and the occluders held into its COB are distributed in the buffers of the newly created elements. In the case M shows a weak interaction between the emitter and receiver, the emitter's radiosity is directly shot to the receiver without using the COB.

```
Element j shoots to element i

if size(i) > SUBDIVISION_THRESHOLD
{
  calculate M = interaction magnitude between j and i
  if M > ILLUMINATION_THRESHOLD
  {
     if COB_i not initialized
        initialize COB_i
     if COB_i is empty
        shoot radiosity from j to i
     else
     {
        subdivide element i
        initialize COBs for all new elements using COB_i
     }
  }
  else
     shoot radiosity from j to i
}
else
   shoot radiosity from j to i
```

Figure 2: Close objects buffer algorithm during progressive refinement

3 Discussion

The COB has proven to be an efficient method of detecting sharp shadows cast by small objects located nearby the surface of large objects, like the paper sheets placed on the desk in Figure 3. It is very likely that the initial meshing of the large object (the desk) will be coarse enough such that it won't be able to capture the small but sharp shadows cast by the objects placed near its surface (the two paper sheets). The COBs of the desk's elements will detect the presence of the close occluding objects and start the subdivision that will ultimately capture the detail shadows.

The COB is not built in a preprocessing stage (as the one used for some shadow casting algorithms) but on demand, only for those elements for which the illumination criteria previously mentioned detect a strong sharp shadow probability. This approach saves a considerable amount of time and space since the COB will be effectively used only for the small number of elements which fulfill the above criteria. The COB initialization method is essentially a lazy evaluation. A similar technique has been used for the evaluation of the links in hierarchical radiosity algorithms [5], [6], based on similar criteria that estimate the interaction strength between elements.

The COB will initiate and guide a mesh refinement up to a user desired precision. Since it works entirely in object space it can generate a very accurate discontinuity mesh, similar to the one produced by shadow casting methods.

The initialization of the COBs can be substantially speeded by using the octrees that some radiosity tools computing form factors based on ray-tracing use for speeding up ray-object intersections [9], [3]. The speed-up is based on the fact that the objects potentially intersecting an element's COB are a subset of the objects contained in the octree cells over which the element spans. Moreover, when an element is subdivided, the buffers of the resulting new elements can be rapidly computed by distributing the objects in the original element's buffer. If the objects are represented by polygons, the COB is a close polygons buffer , and the COB-object intersection test can be implemented as a simple bounding-box-polygon intersection test.

The COB has a storage cost which is linear with the number of elements of the scene since there is one COB per element. Since most of the elements have an empty COB requiring only 5 bytes in the current implementation, the total storage required by the close objects buffers is very small.

It is important to note that the COB has been designed for the detection of sharp shadows cast by *occluders that are close to the occluded objects* and not as a full replacement for the more general discontinuity meshing algorithms based on shadow casting. A radiosity tool that must keep the time, space and implementation costs at a minimum can however use solely the COB in order to detect only the sharp shadows cast by close occluders. Both methods can coexist in the framework of the same radiosity process and perform their best over different areas of the scene.

4 Setting the Parameters

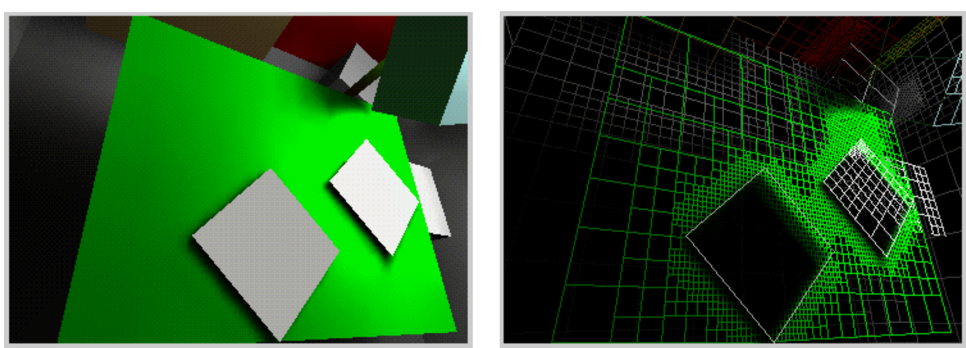
The COB algorithm is influenced by the values of the COB height and the emitter-receiver interaction threshold parameters introduced in Section 2. If the COB height is too small, then objects close enough to the receiver to cast sharp shadows but not close enough to intersect its COB will be skipped by the COB algorithm and the respective shadows will be lost. If the COB height is too large, the COB will include objects which are actually too far from the receiver to cast sharp shadows, and the algorithm will subdivide the receiver uselessly. A good estimate for the COB height is a given fraction of its element's size or some fixed value (e.g. 3% of the environment's size).

Similarly, a too small emitter-receiver interaction threshold will cause a great amount of subdivision, while a large one might fail to detect some visible shadows. A good solution is to compare the emitter-receiver interaction magnitude M with the receiver's actual radiosity B to determine if M will cause a visible change of B and subdivide if M is larger than a small, fixed fraction of B (e.g. 30

5 Examples

Figure 3 shows a detail from an environment rendered with the COB technique. Notice how the sharp shadow created by the sheet of paper placed close to the desk's surface is accurately captured by a fine meshing. Figures 4 and 5 show another interior rendered with the COB algorithm together with the produced mesh.

The time overhead introduced by the COB was less than 2% of the total rendering time. The first iterations of the progressive refinement were visibly slower since most of the elements' COBs were not yet initialized. After a few iterations, most buffers get initialized so the process practically reaches its normal speed.



(a) Rendered image

(b) Discontinuity meshing

Figure 3: A detail view from an office room rendered using discontinuity meshing based on the COB

Figure 4: Office room rendered with the COB algorithm

References

[1] Cohen, M. F., and Wallace, J. R. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Cambridge MA, 1993.

[2] Cohen, M. F., Wallace, J. R., E., C. S., and Greenberg, D. P. *A Progressive Refinement Approach to Fast Radiosity Image Generation*. SIGGRAPH, 1988.

[3] Glassner, A. S. *Space Subdivision for Fast Ray Tracing*, vol. 4. IEEE Computer Graphics and Applications no 10, 15-22, 1984.

[4] Goral, C. M., Torrance, K. E., Greenberg, D. P., and Bataille, B. Modeling the interaction of light between diffuse surfaces. *SIGGRAPH* (1984), 213–222.

[5] Hanrahan, P. M., and Salzman, D. *A Rapid Hierarchical Radiosity Algorithm for Unoccluded Environments*. Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics, 1990.

[6] Hanrahan, P. M., Salzman, D., and Auperle, L. *A Rapid Hierarchical Radiosity Algorithm*. Computer Graphics (SIGGRAPH '91 Proceedings vol 25 no 4), 1991.

[7] Heckbert, P. S. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, University of California, Berkeley, June 1991.

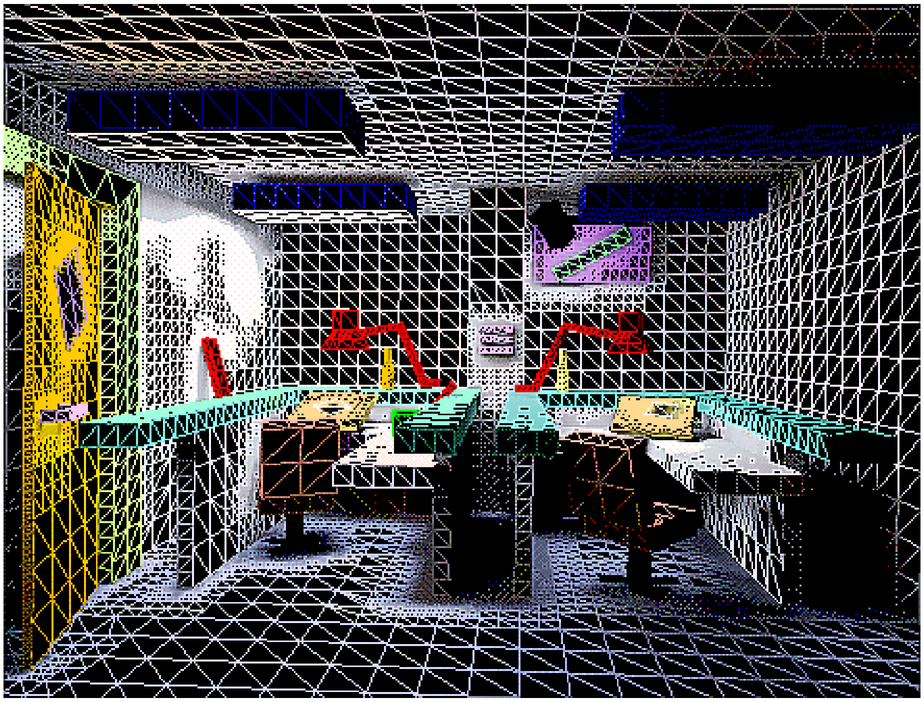


Figure 5: Office room rendered with the COB algorithm (discontinuity mesh)

[8] Nishita, T., and Nakamae, E. *Continuous-Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection*. SIGGRAPH, 1985.

[9] Wallace, J. R., Elmquist, K. A., and Haines, E. A. A ray tracing algorithm for progressive radiosity. *SIGGRAPH 23* (1989), 315–324.