ANTI-ALIASED HEMICUBES FOR RADIOSITY

Sharat Chandran
sharat@cse.iitb.ernet.in

G. Naga Kiran
naga@cse.iitb.ernet.in

M. S. Phani
msphani@usa.net

Chetan Rai
crai@cs.stanford.edu

Computer Science & Engineering Department
Indian Institute of Technology, Bombay
India

ABSTRACT

This paper revisits the classical hemicube and proposes a new form factor algorithm for radiosity solutions. The algorithm substantially reduces aliasing problems inherent in the *hemicube method* without giving up several of the advantages of the hemicube. The algorithm has its basis in the technique of area anti-aliasing in image rendering. Unlike other methods we explicitly consider the effect of the relative order of partial visibility in a hemicube cell when recording form factor computations.

The method can be embedded in several existing radiosity algorithms such as progressive refinement and adaptive substructuring radiosity. The related area anti-aliasing scan conversion algorithm can also be used in applications other than radiosity (such as image rendering of complex scenes on a raster display device).

We provide quantifiable error bounds for our algorithm, and provide experimental empirical results.

Keywords: radiosity, form factor, anti-aliasing, illumination models.

1 INTRODUCTION

The radiosity method [1] is a powerful technique for increased realism in computer generated images. A key procedure which dominates [2], [3], [4] the running time in this method is the computation of form factors.

While closed-form solutions for the double integral that defines the form factor exists, they are too expensive to use. The hemicube method was introduced in [5] as a simple numerical method to compute approximate form factors. Subsequent research (e.g. [6], [7]) pointed out several problems with this method and a ray tracing approach has been advocated. In this paper, we take a fresh look at the classical hemicube method, and propose an anti-aliased hemicube method (AAHemicube).

1.1 PROBLEMS WITH THE CLASSICAL HEMICUBE

There are three principal problems with the classical hemicube method.

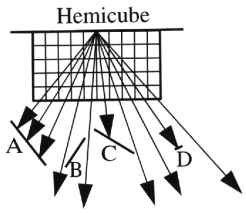

Figure 1: One form of aliasing in the hemicube method due to limited resolution. B is missed.

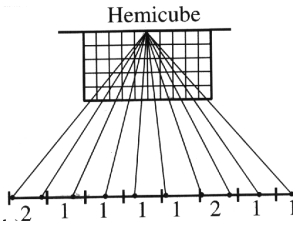

Figure 2: Another form of aliasing. Patches of the same size map to different number of cells.

The first problem relates to the use of a point-to-patch form factor instead of an area-to-area form factor. The second problem relates to the assumption of "constant-visibility" from a source patch to a receiver destination patch across the destination patch. The third problem relates to the limited resolution of the hemicube: patches may be deemed invisible because a ray shot through the center of a cell misses a patch (Figure 1), or an "earlier" patch (in the z-buffer sense) has been allocated the delta form factor [6]. This results in classic aliasing.

Several techniques have been proposed to address the fact that if patches are "large" the first two problems cause poor global illumination computation. Progressive refinement radiosity with adaptive sub-structuring (APR) chop up large patches into tiny ones for the purpose of receiving radiosity. Hierarchical radiosity (HR) [8], [9] with its theoretical roots in wavelets also subdivide patches. Unfortunately, as patches or elements become smaller, the third problem becomes even more troublesome. See Figure 1 and 2. It is well known [2]

that increasing the resolution in the hemicube, or supersampling, is not a workable solution. As a result, in form factor computations in APR and HR, a raycasting based form factor method is used.

1.2 RAYCASTING

The raycasting approach turns around the problem of computing patch-to-patch or patch-to-element form factor on its head. Instead of computing the form factor from a source patch to an element, it computes the form factor from vertices in the scene to the source. As elements become tiny, there is no aliasing effect since only point-to-patch form factors are needed. See Figure 3 (from [2]).

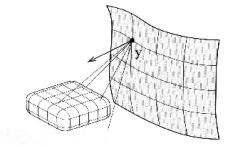

Figure 3: Aliasing is eliminated in the raycasting method.

The raycasting approach especially with Monte Carlo rays has advantages beyond accurate form factor computation (for example, modeling point sources). However, there are disadvantages. It is known [2] that the raycasting approach is computationally expense even discounting the hardware acceleration capability of the hemicube. A more serious problem is the issue of vertex to patch visibility which is obtained by shooting rays. This essentially implies a (potentially uneven) sampling, thus causing an aliasing in the signal processing sense in visibility determination. Finally, the raycasting approach computes patch radiosity by averaging vertex radiosities with weights. This may cause errors if the vertices are bright, but the area "in between" is in a shadow.

1.3 OUR CONTRIBUTIONS

These disadvantages direct us to reconsider the classical hemicube in an anti-aliased setting. Despite the computational advantage, the fact that the hemicube does the twin job of visibility determination, and form factor computation was considered to be a disadvantage primarily because of the aliasing effects. The key idea in our approach, dubbed AAHemicube (Section 3), is to turn this to an advantage by considering patches in relation to other patches when seen from a source patch. This aspect has largely been ignored, with the exception of work in [10] and [11]. AAHemicube produce images that are superior in visual quality to the classical hemicube if the same resolution is used in both cases with a modest increase in time. Alternatively, to match the quality of AAHemicube, a supersampled classical hemicube has to be used with the result that the time taken for rendering in the classical method is substantially higher. Indeed, at least in some cases, for about the same visual quality, AA-Hemicube seems to take less time than HR radiosity without having the unnecessary subdivision [9] of HR radiosity.

It is significant to note that AAHemicube can be incorporated in most radiosity solutions such as progressive refinement (PR) radiosity and APR. It may be noted that APR radiosity with raycasting appears to run significantly slower than wavelet based methods [12] (memory consumption is a different story). Our results [13] seem to indicate that APR radiosity with AA Hemicube take about the same time as HR radiosity, but produces better adaptive meshes.

Another aspect to note in our work is that we provide mathematical error bounds on the amount of errors introduced by our anti-aliased approach.

The rest of this paper is organized as follows. In Section 3 we present an algorithm which reduces aliasing by removing the restriction that a patch must project onto a whole number of pixels and allows pixels to be *partially* covered by patches. The method has its basis in simple area anti-aliasing, a classic signal processing technique that handles aliasing issues by pre-filtering with a low pass filter. We describe this algorithm in Section 2. In Section 4, we provide empirical results based on our implementation to support our approach.

2 ALTERNATIVE ANTIALIASED POLYGON SCAN CONVERSION

Simple scan conversion algorithms consider each pixel as a mathematical point. The conversion is done by sampling at these "points," and aliasing is the result of undersampling. For example, a solid area scan conversion algorithm might base the determination of whether or not a pixel is inside or outside the area on the location of the center of the pixel. This approach can be improved by simple area anti-aliasing which is equivalent [14], [15] to prefiltering with a box function. We describe an efficient algorithm.

```
initialize arrays A and C to zero
for each edge e of the polygon
(* in any order *)
{
   get the sign s of the edge e
   for each pixel (i,j) crossed by edge e {
     A(i,j) + = s (x_e - x_l) (y_e + y_l)/2
     (* all co-ordinates are measured from
     the lower left corner of pixel (i,j).
     Subscripts e and l refer to entering and
     leaving co-ordinates *)
     C(i,j) = C(i,j) + s (x_e - x_l) p_y
     (* p_y is the height of a pixel *)
   }
}
for each column j {
   CC = 0 (* column cumulative carry *)
   for each row i from top to bottom {
     A(i,j) = A(i,j) + CC
     CC = CC + C(i,j)
   }
}
(* A holds the area within each pixel *)
```

Figure 4: Algorithm for area scan conversion

2.1 The Solid Area Scan Conversion Algorithm

The solid area scan conversion algorithm for area anti-aliasing needs to do extra work to determine the area of each pixel within the solid area. This means that, when using an ordered edge list algorithm for polygons, we have to work with a pixel traversal algorithm for the edges instead of a line segment drawing algorithm such as the Bresenham algorithm. Also, this pixel traversal algorithm has to calculate the area under the edge it is traversing; for the interior of the polygon, we know that the entire pixel is covered by the solid area. An improved scan conversion algorithm meant especially for area scan conversion in hemicube radiosity is described in Figure 4.

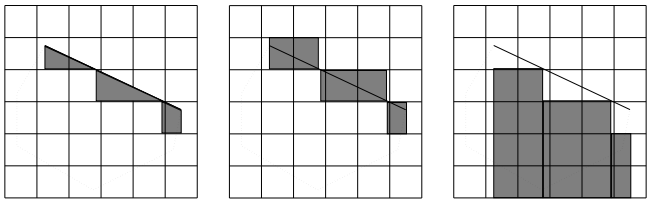

Figure 5: These figures show for one edge of a polygon and for each pixel (a) the area by which A is updated, (b) the area by which C is updated and (c) the effect of C

This method corresponds to calculating the area of a polygon by calculating signed areas under each edge and summing them; the difference is that we need the area *in each pixel*. It works in two phases: *edge traversal* and *carry adjustment*. In the edge traversal phase, we perform a pixel traversal for each non-vertical (with respect to any fixed baseline, such as the bottom of a view rectangle) edge of the polygon. At each pixel (i, j), we add to $A(i, j)$ (the area array) the area under that part of the edge which lies inside the pixel. $C(i, j)$ (the carry array) is updated by the area that would be covered if the edge segment within pixel (i, j) were projected onto the baseline. In the carry adjustment phase, we simply perform a prefix computation for each column using the carry array. The working of the algorithm is illustrated by Figure 5. The superposition of Figure 5 (a) and Figure 5 (c) gives the area under the edge for each pixel.

3 AAHEMICUBE

The following sections describe the proposed radiosity algorithm in the setting of PR radiosity.

3.1 PATCH GENERATION

Our algorithm places no constraints on patch generation; typically, the patches generated are quadrilateral or triangular.

3.2 FORM FACTOR DETERMINATION

As with the original hemicube method, we construct an imaginary unit hemicube around the center of each patch when calculating the form factors from that patch to the other patches in the environment. Before we project an element onto the hemicube surrounding the patch with maximum energy p_{max}, we perform back face culling of the parent patch of element, with the center of patch p_{max} as eye position. If this patch is not hidden,then the element is clipped onto the hemicube and it is area scan converted using the algorithm discussed in section 2.1. Each pixel of the hemicube face contains a sorted list of the *fractional area* A_i covered by the ith element on the pixel. (The size of the list is, in practice, a small subset of the number of elements in the scene). Significantly, we calculate the form factors by taking into account the effects of occlusion by intervening patches as discussed in Section 3.2.1).

3.2.1 ESTIMATION OF AREA COVERED BY AN ELEMENT

The area of a pixel covered by an element is estimated by its expected value. Consider an arbitrary pixel on the hemicube. Let A_0,A_1, A_2, ...,A_{n-1} be the areas of the projection of n ordered patches P_0,P_1,P_2, ...,A_{n-1}. (We adopt the convention that patch 0 is nearest to the pixel).

Let f_i be the fraction of form factor of the pixel which has been allocated to patches 0 through i.

The method estimates the area of the visible portion of patch k, namely E_k, by

$$\begin{aligned} E_k &= A_0 \ \ for\ k = 0 \\ &= A_k(1 - f_{k-1}) \ \ for\ 0 < k < n \end{aligned}$$

In other words, we inductively know the area covered by an element without considering occlusion, say A_k, and the fraction f_k, which is the area already accounted for in a pixel. The probability that an infinitesimal area will not be occluded by elements already processed given that it projects onto the pixel is the fraction of the area of the pixel not covered by those patches.

The sorting of the elements is done dynamically as in insertion sort. When the entire pixel area is consumed, no items are inserted in the list corresponding to a pixel. The scan conversion happens in any case. (The procedure is modified slightly when elements belong to the same physical surface indicated by identical z-values.)

3.2.2 ANALYSIS OF ERRORS

In this section, we formally analyze the error in the proposed method. We start with computing a closed form expression for the form factors assigned to each patch.

3.2.3 CLOSED FORM EXPRESSION

Note that E_k is also the fraction of the form factor of the pixel that is allocated to patch k. Thus, $f_0 = E_0 = A_0$ and $f_k = f_{k-1} + E_k \ \ 0 < k < n$. Solving the coupled equations results in the closed form expression, $E_1 = A_1(1 - A_0)$ and $E_k = A_k(\prod_{i=0}^{k-1}(1 - A_i)) \qquad k > 0$

3.2.4 DEFINITIONS

For the purpose of analysis, we divide possible input into three cases. We say that a patch is *underestimated* (respectively, *overestimated*) when the form factor assigned to it by our algorithm is less (respectively, more) than the true form factor. An input set of patches is said to belong to the *pure underestimated* variety when every patch is visible. An input set of patches is said to belong to the *pure overestimated* variety when every patch (except the first) is occluded. Otherwise, an input set is said to belong to the *mixture* variety.

3.2.5 PURE UNDERESTIMATION

In [16] we show that the worst case underestimation error expression is $(1 - 1/n)^n$ which tends to $e^{-1} \approx 0.37$ as n tends to infinity.

3.2.6 PURE OVERESTIMATION

In [16] we show that the the expression for the total error is maximized when $A_0 = 1 - (1/(n-1))^{\frac{1}{n-2}}$ in which case, the maximum error is

$$\left(\frac{1}{n}\right)^{\frac{1}{n-1}} - \left(\frac{1}{n}\right)^{\frac{n}{n-1}}$$

3.2.7 MIXTURE CASE

The sum of absolute errors when both kinds of errors, underestimation and overestimation, are present may be significant. But under this case, the presence of both kinds of errors, has an averaging effect on the form factor errors rather than a cumulative effect as in the previous cases. In the average case this error is zero. This result is derived in [16].

4 RESULTS

In this section we consider the proposed algorithm in the context of PR radiosity using the classical hemicube (without adaptive substructuring), PR radiosity using raycasting (without adaptive substructuring), and HR radiosity using raycasting.

The test cases highlight the two main problems in the hemicube method. The first case identifies the visual aliasing effects due to discretization of surface into polygons such that their projection on hemicube nearly matches the spacing of hemicube cells. The second case considers a more serious problem. Small patches (patches on the circular table) cover less than one cell and is missed entirely by the classical method, unless supersampling is employed.

The test code uses the software [4] (which we modified for the purposes of uniformity) for computation of patch radiosities. In order to compare the two methods, the code for both algorithms is kept the same, except for the calculation of formfactors. We have compared our method with the original method with the same parameters for convergence. The convergence criterion is set to 1% of the total unsent energy. This means that when the unsent flux becomes less than 1% of the total flux, we say that convergence has reached.

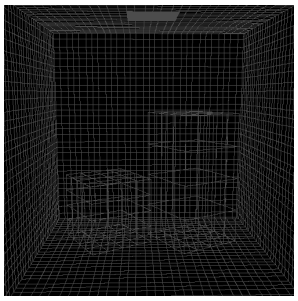

Figure 6: The first test scene.

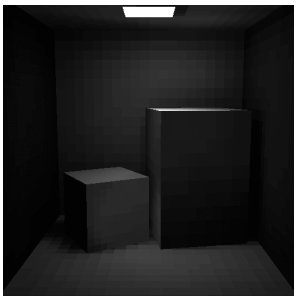

Figure 7: The proposed method

Figure 8: The classical method

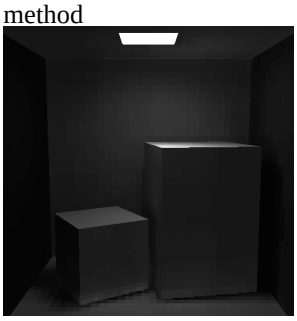

Figure 9: Hierarchical radiosity [8]

Figure 10: Figures 7, 8 and 9 use radiosity values as is, and do not interpolate. The pictures are best viewed in color on a multilevel display.

For the first case, we consider the *Cornell room* as shown in Figure 6. For the test environment we worked with, we observed that a hemicube resolution of 100 is sufficient for our method. In order to compare our method with the classical method, we use the same hemicube resolution. The results (Figure 7 and Figure 8) indicate that our method has eliminated the aliasing effects whereas the original method shows considerable aliasing. All figures except when mentioned otherwise explicitly have been rendered with only the radiosity values, and not been interpolated. No figure uses any hardware acceleration features. Our results were generated in 10 seconds as opposed to 6 seconds for PR radiosity with the classical hemicube.

We also compared our method with our implementation of HR radiosity For this test, the input patches for the hierarchical method are at their coarsest level. The results show that the two images (Figure 7 and Figure 9) are visually comparable Our results were generated in 10 seconds as opposed to 21 seconds for HR radiosity.

We also compared the implementation of raycasting with PR radiosity. The radiosity values and the resulting bilinear interpolated image are shown in Figures 11 and 12. Our results of Figure 7 were generated in 10 seconds as opposed to 159 seconds for the raycasting approach of Figure 11.

For a second test case, we consider the situation in Figure 13. This scene consists of a closed room with three office chairs and a pedestal table. The scene consists of a total of 12 surfaces which comprise of 5591 patches and 15784 elements.

For the classical method, we used hemicube resolutions of

Figure 11: Radiosity with the raycasting method using 8 rays.

Figure 12: Bilinear interpolation (raycasting method).

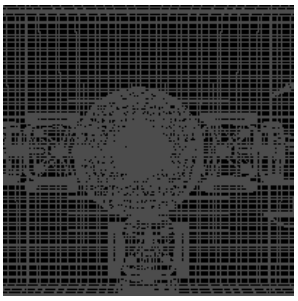

Figure 13: A second test scene

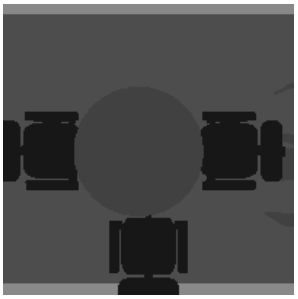

Figure 14: Shades employed.

Figure 15: Patch radiosities obtained by the classical hemicube method (resolution 300)

Figure 16: Patch radiosities obtained by the classical hemicube method (resolution 1200)

Figure 17: Patch radiosities obtained by the classical hemicube method (resolution 2500)

Figure 18: Patch radiosities obtained using the proposed method with hemicube resolution 100

$300, 1200, 2500$ and compared the visual quality of the rendered image with our method (Figure 18 as shown in Figures 15,16,17,18. The time taken for calculating the solution for each of the cases in Figures 15,16,17 are $171s, 432s, 1350s$ respectively, compared to $360s$ in the proposed method.

We observe that the visual quality of the images tend to improve with the hemicube resolution and tend to converge to image obtained by our method. The aliasing effects on floor disappear at a hemicube resolution of 600 in the classical method; however aliasing effects in the center of the table are visible at this resolution particularly if we zoom into this area.

5 CONCLUDING REMARKS

The radiosity algorithm discussed in this paper substantially reduces the aliasing inherent in the classical hemicube method for form factor and visibility determination. We have provided a formal mathematical analysis for the reduction in the aliasing effects in our algorithm. In terms of time, an extra computational cost is incurred due to the fact that we need to calculate areas covered during scan conversion of polygons and accounting for occlusion takes more time than a simple comparison in case of the z-buffer in the original hemicube method. We have also compared our algorithms to radiosity algorithms that does not use the hemicube in any form, and a super sampled classical hemicube algorithm, and shown the extra time is not substantial.

As a precursor to a new form factor algorithm, we have proposed a two dimensional area anti-aliasing algorithm. An earlier algorithm which used area anti-aliasing in the three dimensional image rendering context was proposed in [17]. This algorithm is computationally expensive because it required the determination of actual areas of pixels covered by objects in the image and does not appear to be practical in radiosity. Our two-dimensional algorithm introduced for radiosity purposes may be useful elsewhere.

Our algorithm does not eliminate problems other than aliasing in the hemicube method. It does not have the flexibility of Monte Carlo ray casting, or the adaptive nature of wavelet radiosity which elegantly sidesteps the issue of computing all pairs form factors. It does have the inherent advantage of the hemicube (speed), without consuming too much extra memory.

We have included our algorithm in adaptive substructuring progressive refinement radiosity [13] with promising results noting that the classical hemicube does not work very well in such cases. We believe our work can be embedded in other sophisticated radiosity techniques as well.

ACKNOWLEDGEMENTS

We appreciate the help of Nilesh Dalvi and Ashwin Bharambe ({nilesh,ashu}@cse.iitb.ernet.in).

REFERENCES

[1] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile, “Modeling the interaction of light between

diffuse surfaces," *Computer Graphics*, vol. 18, no. 3, pp. 213 – 222, 1984.

[2] F. Sillion and C. Puech, *Radiosity and Global Illumination*. Morgan Kaufman, 1994.

[3] M. Cohen and J. Wallace, *Radiosity and Realistic Image Synthesis*. Academic Press, 1993.

[4] I. Ashdown, *Radiosity: A Programmer's Perspective*. Wiley Eastern, 1994.

[5] M. F. Cohen and D. P. Greenberg, "The hemicube: A radiosity solution for complex environments," *Computer Graphics*, vol. 19, no. 3, pp. 31 – 40, 1985.

[6] J. R. Wallace, K. A. Elmquist, and E. A. Haines, "A ray tracing algorithm for progressive radiosity," *Computer Graphics*, vol. 23, no. 3, pp. 315 – 324, 1989.

[7] D. R. Baum, H. E. Rushmeier, and J. M. Winget, "Improving radiosity solutions through the use of analytically determined form factors," *Computer Graphics*, vol. 23, no. 3, pp. 325 – 334, 1989.

[8] P.Hanrahan, D.Salzman, and L.Aupperle, "A rapid hierarchical radiosity algorithm," *Computer Graphics*, vol. 25, pp. 197–206, July 1991.

[9] P. Schroeder, "Wavelet radiosity," in *Siggraph 95 Course Notes*, 1995.

[10] F. X. Sillion and C. Puech, "A general two-pass method integrating specular and diffuse reflection," in *Computer Graphics (SIGGRAPH '89 Proceedings)* (J. Lane, ed.), vol. 23, pp. 335–344, July 1989.

[11] R. J. Recker, D. W. George, and D. P. Greenberg, "Acceleration techniques for progressive refinement radiosity," in *Computer Graphics (1990 Symposium on Interactive 3D Graphics)* (R. Riesenfeld and C. Sequin, eds.), vol. 24, pp. 59–66, Mar. 1990.

[12] A. J.Willmott and P. S. Heckbert, "An empirical comparision of progressive and wavelet radiosity," *Eurographics Workshop on Rendering*, 1997.

[13] N. Kiran, S. Mudur, and S. Chandran, "Anti-aliased hemicubes for performance improvement in progressive refinement radiosity," tech. rep., http://www.cse.iitb.ernet.in/ ˜ graphics/nagarad.html Computer Science and Engineering Department, IIT Bombay, 1999.

[14] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques: Theory and Practice*. Addison-Wesley Publishing Company, 1994.

[15] D. F. Rogers, *Procedural Elements for Computer Graphics*. McGraw-Hill Book Company, 1985.

[16] S. Chandran, N. Kiran, M. Phani, and C. Rai, "Anti-aliased hemicube for radiosity computations," tech. rep., http://www.cse.iitb.ernet.in/ ˜ graphics/nagarad.html Computer Science and Engineering Department, IIT Bombay, 1998.

[17] E. Catmull, "A hidden surface algorithm with anti-aliasing," *Computer Graphics*, vol. 12, no. 3, 1978.