ANTI-ALIASED HEMICUBES FOR PERFORMANCE IMPROVEMENT IN RADIOSITY SOLUTIONS

Naga Kiran   S. P. Mudur   Sharat Chandran   Nilesh Dalvi

National Center for Software Technology
Mumbai, India
`mudur@ncst.ernet.in`

Indian Institute of Technology
Mumbai, India
`{naga,sharat,nilesh}@cse.iitb.ernet.in`

ABSTRACT

Several important and fascinating aspects of realistic images are captured by the radiosity method. In this paper we use an alternate form of the classical hemicube that reduces aliasing problems inherent in the original method without giving up the *computational advantages* of the hemicube. Unlike other methods, we explicitly consider the effect of the relative order of partial visibility in a hemicube cell when recording form factors. This enables us to compute form factors accurately (even) in progressive refinement radiosity with adaptive substructuring. Our empirical results with progressive refinement radiosity show superior mesh density where fine details are required, (such as in soft shadows), as well as in areas that produce singularities (such as when inter element distances tend to zero). Our method is contrasted with hierarchical radiosity which uses raycasting for form factor and visibility computations. On a low end Intel Linux platform, and for a comparable image quality, our method takes substantially less time.

Keywords: Anti-aliasing, visibility order, adaptive substructuring,

1 INTRODUCTION

Hierarchical radiosity (HR) [1] and progressive radiosity (PR) with adaptive substructuring [2] are two approaches used in solving the radiosity equation effectively. The beauty of both methods is that a scene is started out with a coarse set of patches, and the algorithm adaptively subdivides the scene into meshes. As a result, undue effort is not spent in radiosity computation when it is not required (for example an area in shadow (the umbra) may not receive light at all). In recent years, considerable work has been done in extending the adaptive techniques, both in terms of efficiency and quality (for example, clustering techniques in [3]), and stochastic methods as in [4], [5], [6], [7]). In an empirical comparison [8], it has been pointed out that the HR methods, with variations, perform extremely well in terms of time. This makes HR methods as the most promising way to perform radiosity computations.

In this context, there are two factors that we study in this paper.

1. **Mesh density:** HR methods require from the user a form factor tolerance factor based on which $O(m)$ interactions are set up, instead of the $O(nm)$ factor in PR radiosity. Here, roughly speaking, m refers to the number of receiver elements, and n refers to the number of shooter patches. It has been observed (for example, in [9]) that HR radiosity introduces, *in addition*, an area tolerance factor. The area tolerance factor is not necessary in the theoretical presentation, but becomes important in the practical implementation because of the singularity when inter-element distances are very small. If, in the interests of accuracy, the factor is set to too small a number, unnecessary subdivisions (typically in corners) and increased computational time are observed.

   A big advantage of the HR radiosity method is that it does *not* consider all possible (quadratic) interactions simply because of the introduction of many elements. At the same time, it does not guarantee radiosity continuity between adjacent elements. While this is true for most radiosity algorithms, the effect of this is visually not pleasing particularly when there are unnecessary subdivisions.

   PR radiosity sidesteps singular situations by computing element-vertices to patch-area form factors; in HR radiosity, there is intentionally no difference between patches and elements. The principal disadvantage of PR radiosity appears to be the computational cost, and the theoretical quadratic complexity as opposed to the linear (in time) complexity of HR.

2. **Occlusion:** As [10] state, determining occlusion is often the most computationally demanding aspect of form factor computation. Changes in visibility can cause perceptually important discontinuities in the radiosity function and must be determined accurately. Unlike other operations in computer graphics such as clipping, and rigid body transformation, visibility between two elements depend on their relative configuration with other surfaces. Therefore, although the total number of interactions is $O(m)$, the process of obtaining the links between elements or groups of elements is a superlinear process. A naive process has $O(mn)$ complexity, as observed in [1]; the sophisticated process suggested in requires "in principle" [1] a theoretical $O(m \log n)$ complexity.

   It is well known that because the hemicube handles visibility and form factor computation together, hemicube-based algorithms run fast. Unfortunately, an attempt to use the hemicube based algorithm for form factor computation in the PR approach fails as can be seen in Figure 1 and Figure 2.

   As a result, the form factor and visibility computation are decoupled in PR radiosity as well as in HR radiosity, and performed using a ray casting approach [11]. The quality, and cost, of visibility computation depends on the number and location of rays fired.

1.1 Our contributions

In this paper, we revisit the hemicube in its anti-aliased [12] avatar in order to leverage upon its computational advantages.

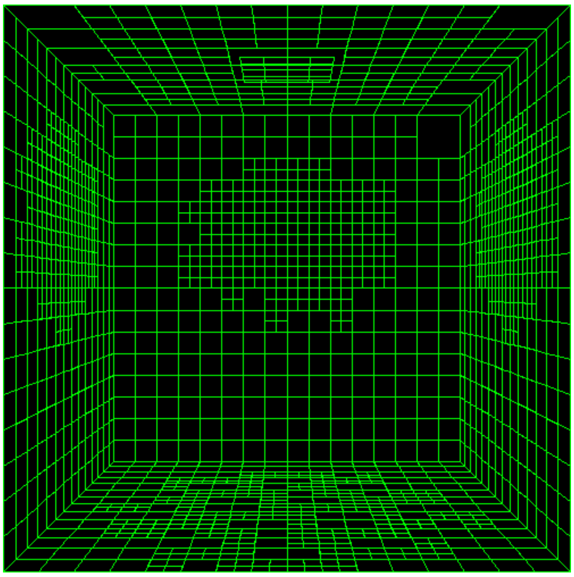

Figure 1: Adaptive meshing using the classical hemicube in progressive radiosity.

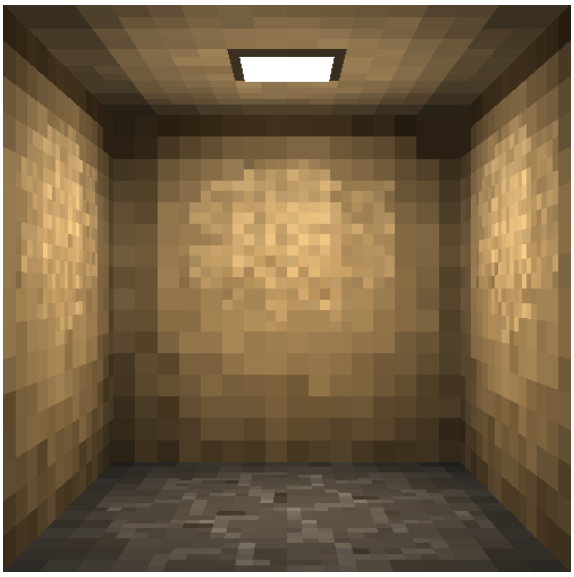

Figure 2: A flat shaded radiosity rendering for the simple room of Figure 1.) with a hemicube resolution of 100.

- **Mesh Density:** The number of meshes in our algorithm is controlled by the error in radiosity computations, which in turn is driven by a more accurate form factor calculation in the anti-aliased setting. Our results show superior meshing in areas of soft shadows thereby preventing display artifacts in such areas, while maintaining the hard-shadow advantage [8] of PR radiosity. An example image is shown in Figure 3.

- **Computational time:** As expected, our algorithm runs fast. Some experimental results are mentioned in Section 3.

Although we have implemented the anti-aliased hemicube (AAHemicube) only in PR radiosity, there is no restriction on it being used in (a modified form) in HR radiosity in the computation of form factors.

The rest of this paper is organized as follows. In the next section a brief description of the AAHemicube algorithm is given. In Section 3, we provide empirical results based on our implementation to support our approach. Some concluding remarks are made in Section 4

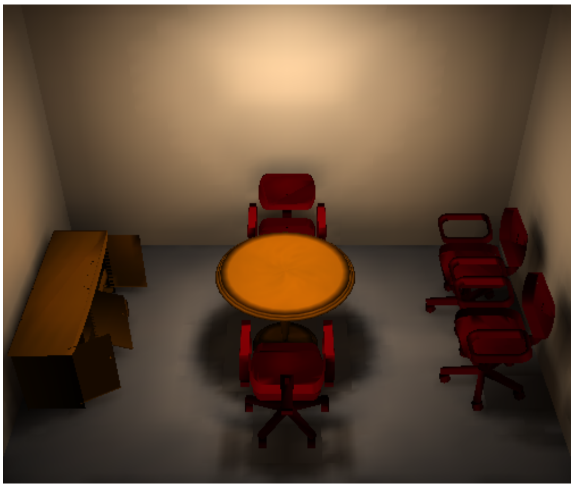

Figure 3: The interrogation chamber.

2 ANTI-ALIASED HEMICUBE

Form factor computations form an important part of radiosity calculations. Discussion of the form factor usually involve the solid angle as seen, say, from the center of one patch, and the distance between two patch centers. In addition, though, the visibility of a patch from other patch is also an important aspect of the form factor computation. Ray tracing methods use different approaches to solve the two problems. The classical hemicube [13] performs both tasks together which largely accounts for the speed of computation. In addition, the hemicube is amenable to the benefits of hardware z-buffering.

Its advantages notwithstanding, the classical hemicube suffers from the problems of aliasing [14]. This is due to the regular sampling of space into cells by an invisible cube (hence the name) reminiscent of pixels in a CRT display. As a result, when elements become small as in PR radiosity (or HR radiosity), they "slip-through-the-cracks". This results in patches not receiving any light, as can be seen in Figure 2. Further, a stipulation is made that only one element (or patch) can receive energy through a cell of the cube. This results in some patches receiving more energy than necessary, an effect that can also be seen in Figure 2. This is one reason for the plaid-like effect seen in older, hemicube driven radiosity rendered images.

AAhemicube takes care of the aliasing problem by pre-filtering the cells by a low pass filter. In practice, this is achieved by removing the restriction that a patch or element must project onto a whole number of cells, and allows cells to be partially covered by elements or patches. There are two steps needed in the process. We describe this briefly in Section 2.1 and Section 2.2 (more details are available in [12]).

2.1 The Solid Area Scan Conversion Algorithm

First, a method is needed that will scan convert polygons, and determine the area of each hemicube pixel within the polygon. An algorithm for this purpose is described in Figure 4. (It is also useful, with slight modifications, as a fast, generic polygon scan conversion 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 $j$
{
  (* initialize column cumulative
  carry *)
  $CC$ = 0
  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: An algorithm for fractional area computation.

2.2 Form Factor Determination

As with the original hemicube method, we construct an imaginary unit hemicube around the center of the patch p_{max} with maximum energy to be shot. Before we project an element onto the hemicube we perform back face culling of the parent patch of element, with the center of patch as eye position. If the patch is not hidden, then the element is clipped onto the hemicube and it is area scan converted using the method of Figure 4. Thus all relevant elements (and thus patches) are projected on to the hemicube as opposed to only one in the classical hemicube. 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 explained below.

2.2.1 Fractional form factors

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, $\ldots$,A_{n-1} be the areas of the projection of n ordered patches P_0,P_1,P_2, $\ldots$,P_{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 the equations,

$$E_0 = f_0 = A_0$$
$$E_k = A_k(1 - f_{k-1}), \text{ for } 0 < k < n$$
$$f_k = f_{k-1} + E_k, \text{ for } 0 < k < n$$

2.3 Adaptive mesh refinement

The process of adaptive subdivision takes place in much the standard way [10]. The contribution of the direct illumination by the current emitter is isolated. Next, the *a posterio* subdivision criterion based on the form factor that has been computed along with the visibility estimate is used to decide which elements are too large. The key advantage of the proposed method is that when an element is subdivided, the visibility estimate of the parent element (or patch) is used in computing the visibility estimate of the child element. (This is possible because the pixels on the hemicube that any child projects to is a subset of the original set of pixels covered by the parent.) The form factor is computed using this visibility estimate. All elements that are found to require subdivision are split, and the current “shot” is effectively cancelled. A new shot is then performed from the same emitter using the new set of elements. This may result in a new set of subdivisions. At some stage, subdivision stops, and a step of the Southwell relaxation is performed. The energy due to the current emitter is then interpolated with the previous energy values of the former elements.

3 EXPERIMENTAL RESULTS

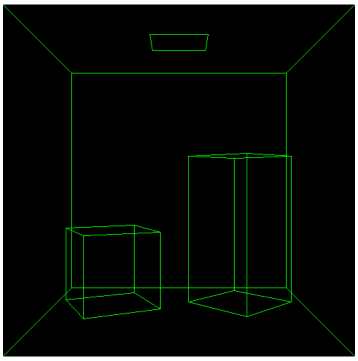

Figure 5: Wireframe image of the Cornell room

Using AAHemicube on the environment in Figure 5, a picture of the Cornell room

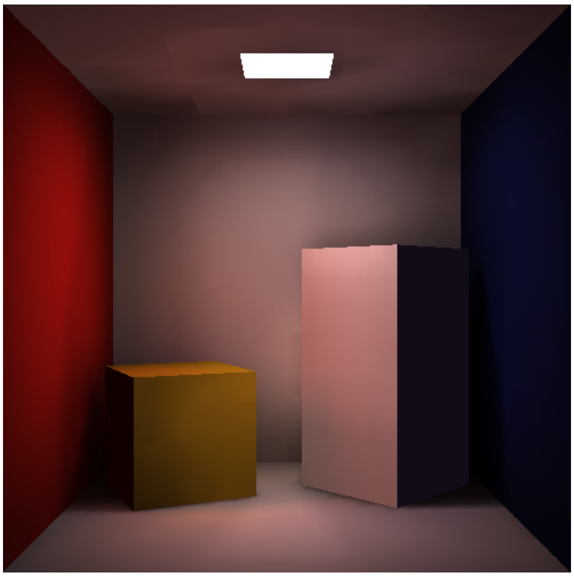

Figure 6: Rendering using the proposed method (hemicube resolution 100 and AEPS=0.0001 and FEPS=0.001).

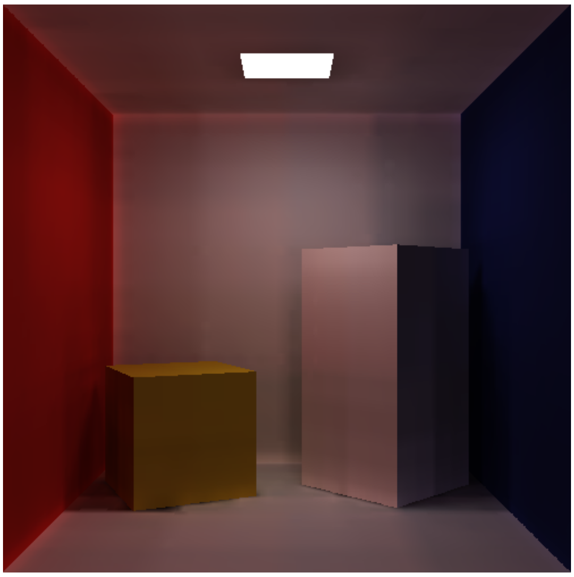

Figure 7: Rendering using hierarchical radiosity (AEPS=0.0001 and FEPS=0.001). Observe the bright areas in room corners.

rendered (without interpolation) is shown in Figure 6. Figure 7 shows the same scene rendered with the same form factor tolerance (FEPS) and area tolerance factor (AEPS) using the HR method[1]. At first glance, both scenes appear equally pleasing[2] in their own way (these figures, and others, are best seen on a computer workstation with interactive zooming capability). It is interesting to note that the time taken is 20 seconds for the proposed method, and 98 seconds for the HR method on a low end Intel Linux platform.

We examine the two picture more closely. Figure 8 and Figure 9 show the meshing performed on the Cornell room by the two methods. There are two places of interest: one is the corner of the room (wall and floor), and the other is the interaction between the smaller and larger blocks.

At corners, the HR recursive procedure performs considerable subdivisions, and finally terminates because of the area tolerance factor instead of the form factor tolerance factor. As a result, an overestimation of the true radiosity is made, and corners appear brighter. A closer examination of the HR picture also shows that the inter reflections between the smaller block and the larger block has not been rendered accurately.

The solution is to reduce the area tolerance factor further. We observed that when we set the area tolerance factor in the HR method to a smaller value, HR takes an inordinate amount of time to converge on our low end workstation. In contrast, since form factors are computed more accurately using AAHemicube, we find that a less stringent form factor tolerance and area tolerance factor is sufficient to converge to a visually pleasing solution.

We have also experimented our algorithm on a more complex scene containing 6276 patches. The input wireframe model of the environment is shown in Figure 10. The final meshing is shown in Figure 11.

[1] Code obtained from the `papers/rad/code` subdirectory at `http://graphics.stanford.edu`.

[2] A theoretical error analysis has been done in [12].

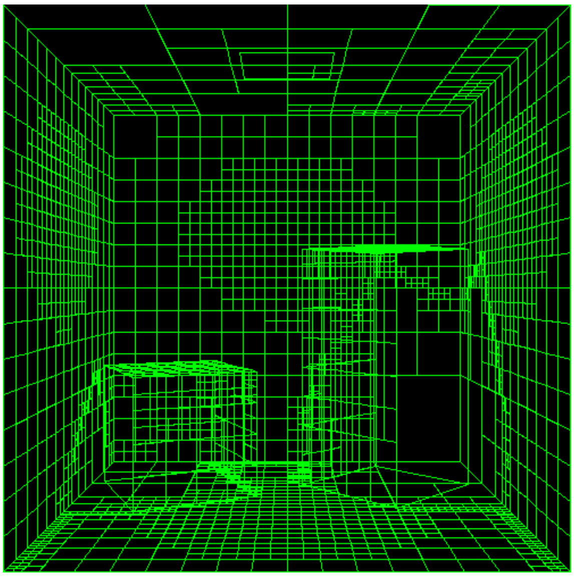

Figure 8: Wire mesh of the rendered image using AAHemicube.

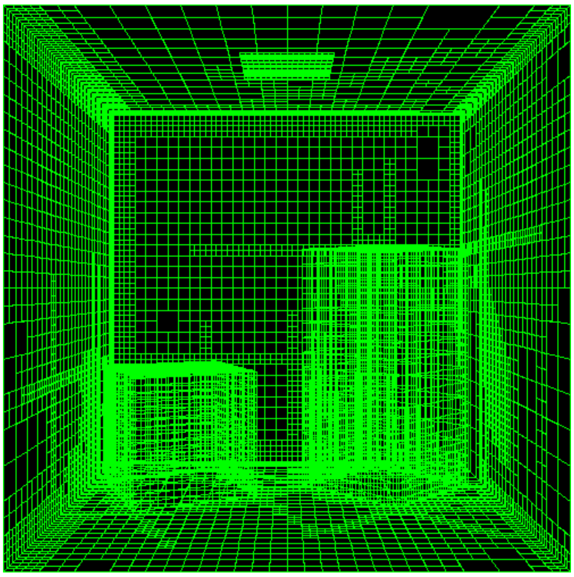

Figure 9: Wire mesh of the rendered image using the HR method.

The bilinear interpolated of Figure 11 is shown in Figure 3. We see that the shadows (on the floor and the walls) are accurately captured. Our current implementation, without the use of any clustering technique, takes 74 seconds to reach convergence with AAHemicube. It is well known [9] that HR radiosity has quadratic complexity if the initial input contains a large number of small patches, as is the case in this situation.

4 CONCLUDING REMARKS

In this paper, we have used the anti-aliased hemicube and applied it to adaptive substructuring problem. A key aspect in this is the reuse of visibility information when an element is subdivided.

The meshing of the rendered scene appears to be satisfactory especially in regions that might result in singularities (distance between elements tend to zero), and regions in soft shadows. At the same time, leveraging upon the advantages of the hemicube, the computational time is significantly small. Also, the use of clustering techniques ([3]) facilitates further increase in the efficiency. We conclude therefore that in the situations where progressive radiosity is to be used (as suggested in [8]), one may consider AA-Hemicube for form factor computations to improve the overall performance.

Several future directions suggests itself. The current implementation uses regular subdivision of surfaces. Since we are able to determine the visibilities of the receiver patches in the environment nearly accurately, one direction is to use this information for irregular subdivision of receiver patches. We also need to experiment the use of the algorithm with other radiosity techniques particularly, hierarchical radiosity.

REFERENCES

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

[2] M. Cohen, S. E. Chen, J. Wallace, and D. P. Greenberg, “A progressive refinement approach to fast radiosity

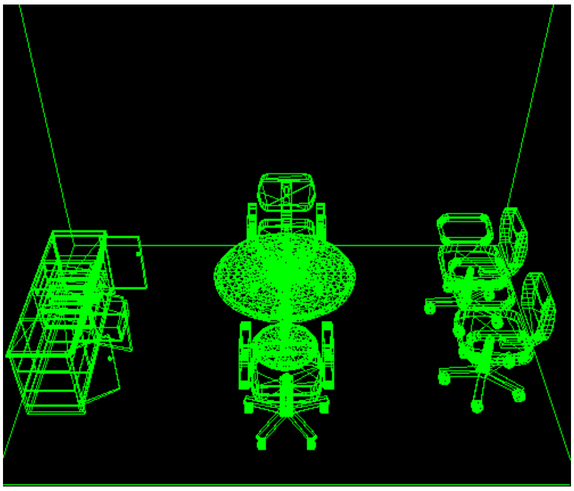

Figure 10: Initial wire frame image of the interrogation room consisting of 6300 patches.

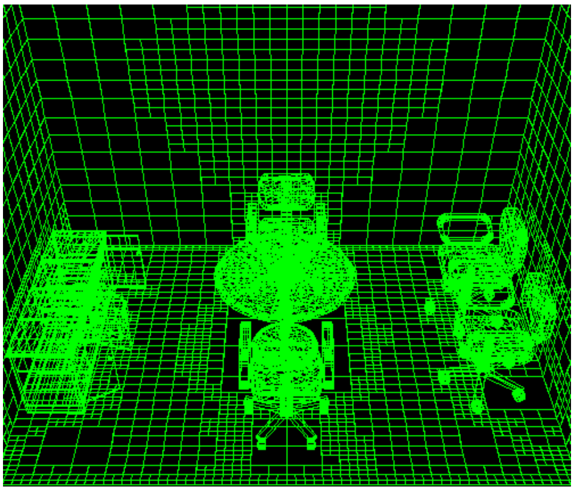

Figure 11: Final wire frame image of the interrogation room.

image generation," *Computer Graphics*, vol. 22, pp. 75 – 84, 1988.

[3] A. Willmott, P. Heckbert, and M. Garland, "Face cluster radiosity," *Eurographics Workshop on Rendering*, pp. 293–304, June 1999.

[4] J. Prikryl, P. Beakert, and W. Purgathofer, "Importance-driven hierarchical stochastic ray radiosity," *7th Eurographics Workshop on Rendering*, June 1996.

[5] R. Tobler, M. Feda, W. Purgathofer, and A. Wilkie, "A hierarchical subdivision algorithm for stochastic radiosity methods," *8th Eurographics Workshop on Rendering*, pp. 193–204, June 1997.

[6] P. Bekaert, L. Neumann, A. Neumann, M. Sbert, and Y. Willems, "Hierarchical Monte Carlo radiosity," *Eurographics Workshop on Rendering*, pp. 259–268, 1998.

[7] J. Pope and A. Chalmers, "Improving hierarchical Monte Carlo radiosity algorithms," *WSCG*, 2000.

[8] A. Willmott and P. Heckbert, "An empirical comparision of progressive and wavelet radiosity," *Eurographics Workshop on Rendering*, pp. 175–186, 1997.

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

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

[11] J. Wallace, K. A. Elmquist, and E. A. Haines, "A ray tracing algorithm for progressive radiosity," *Computer Graphics*, vol. 23, pp. 315 – 324, 1989.

[12] S. Chandran, N. Kiran, M. S. Phani, and C. Rai, "Anti-aliased hemicube for radiosity," *Computer Graphics and Imaging*, 2000.

[13] M. Cohen and D. P. Greenberg, "The hemicube: A radiosity solution for complex environments," *Computer Graphics*, vol. 19, pp. 31 – 40, 1985.

[14] 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, pp. 325 – 334, 1989.