Time Complexity of Monte Carlo Radiosity

Computers and Graphics 16(1) 1992 (originally in Eurographics '91)

Peter Shirley *
Department of Computer Science
Indiana University

Abstract

The time complexity of Monte Carlo radiosity is discussed, and a proof is given that the expected number of rays required to produce a statistical radiosity solution below a specified variance for N zones is $O(N)$. A satisfactory solution is defined to be one in which the variance of radiance estimates for each zone is below a predefined threshold. The proof assumes that the radiance is bounded, and the area ratio of the largest to smallest zone is bounded.

1 Introduction

In a radiosity (zonal) program, the surfaces in the environment are broken into N zones, z_i, and the radiance, L_i, of each zone is calculated [6, 7]. In the most straightforward radiosity method, all N^2 relationships (form-factors) are explicitly calculated, so the time complexity of the program is *at least* $O(N^2)$.

One of the first schemes to lower the radiosity calculation time was to group the N zones into p patches, and transfer power from patches to zones (elements) [4]. Still, the computation time will be at least $O(Np)$, so many zones need to be grouped before a large savings occurs.

Hanrahan and Salzmann [8] observed that there are similarities between the N zone radiosity problem and the N body gravitational simulation problem. In the gravitation problem, all N^2 relationships must be established to calculate the accelerations of the bodies. Calculating all relationships can be avoided by calculating acceleration *to within a specified error tolerance*. Because some error is then acceptable, the relationships between groups of objects can be calculated, and the problem can be solved in $O(N)$ time. Hanrahan and Salzmann used similar techniques to solve the radiosity problem (without shadowing) to within a specified error tolerance in $O(N)$ time. Hanrahan et al. extended this work to shadowed environments by adding a ray tracing shadow test, and showed that this required a total of $O(N)$ rays [9].

One potential drawback of the two optimization techniques described above, is that they rely on the ability to group zones into larger patches. In a highly complex scene this may be nontrivial.

*Department of Computer Science, Lindley Hall, Indiana University, Bloomington, IN 47405. Email: shirley@cs.indiana.edu.

One method that does not rely on such an ability is a Monte Carlo simulation [11, 12, 1, 13, 10]. The Monte Carlo method can be applied to gravitational simulation by probabilistically choosing which of the N^2 relationships to calculate, and using these as estimates of the uncalculated relationships. This will not yield a solution within a known error tolerance, but will yield a solution within a *specified variance*.

In a Monte Carlo radiosity program, each zone sends power to other zones by sending rays in the directions that power is radiated. Details of this approach can be found in [14]. Both Airey [2] and this author [14] have observed that, in practice, only $O(N)$ rays seem to be required for an N zone solution.

In this paper, the theoretical time complexity of the Monte Carlo radiosity method is investigated, and a proof is given that the expected number of rays required to come within a specified variance is $O(N)$, given that the environment satisfies certain constraints. Assuming it takes $O(\log N)$ time to trace a ray, that implies an $O(N \log N)$ total solution time.

2 Direct Lighting

Suppose we are generating radiance values for a particular diffuse scene that has been discretized into zones. Assume that we are going to use Monte Carlo ray tracing to estimate the radiance values. How many rays are needed for a satisfactory solution?

To begin an attack on the problem, assume that we only want to estimate the direct lighting (the light that come directly from luminaires to zones).

Rays will be independently emitted from light sources, each carrying the same amount of power (each of r rays will carry Φ/r power, where Φ is the total power). What we want to estimate is the radiance, L_i of each zone. We want to make r large enough so that the variance of our estimate for all L_i is below a specified threshold, V_0.

Because we are sending *power* carrying rays, it is convenient to calculate the power, Φ_i, reflected from z_i. By definition, the radiance of z_i can be related to the power leaving it:

$$L_i = \frac{\Phi_i}{\pi A_i}$$

where A_i is the area of z_i.

Before trying to figure out how big r must be, let's establish some useful relations. Suppose we have a sum, S, of N independent identically distributed random variables X_i, where each X_i is a value x with probability p and zero otherwise. We can immediately establish:

$$E(S) = E(\sum_{i=1}^{N} X_i) = N E(X_i) = Npx \tag{1}$$

and the variance of S is:

$$var(S) = var(\sum_{i=1}^{N} X_i) = N\, var(X_i)$$

$$var(X_i) = (E(X_i^2) - E(X_i)^2) = (px^2 - p^2x^2) \leq px^2$$

So we can establish the inequality:

$$var(S) \leq Npx^2 \tag{2}$$

Initially we send r rays from the emitting zones. We want r large enough so that the variance in our estimate for every L_i, $var(L_i)$, is below some predefined threshold V_0. The reflected power from the ith zone is:

$$\Phi_i = R_i \sum_{j=1}^{r} \phi_i^j$$

where ϕ_i^j is the amount of power z_i receives from the the jth ray. Since each ray ihits exactly one zone, the ith zone will either get all of its power (probability p), or none of it (probability $1-p$). Because all rays are generated independently and according to the same distribution, p depends only on i, and not j, and will thus be denoted p_i. This means the expected value of Φ_i is (from (1)):

$$E(\Phi_i) = R_i p_i \Phi$$

So the expected radiance of the ith zone is:

$$E(L_i) = \frac{R_i}{\pi A_i} p_i \Phi$$

Similarly, from (2) the variance of the power estimate satisfies the inequality:

$$var(\Phi_i) \leq R_i^2 p_i \frac{\Phi^2}{r}$$

and similarly:

$$var(L_i) \leq \frac{R_i^2}{\pi^2 A_i^2} p_i \frac{\Phi^2}{r}$$

To ensure that $var(L_i) < V_0$, we may have to make r very large. An obvious problem occurs if $A_i = 0$. This will cause unbounded variance, which is not surprising; a point has a radiance, but estimating the radiance of a point does not fit into our scheme. We could simply assume a lower bound for A_i, but for some N this lower bound would become impossible. If the total surface area in the scene is A, then there must be some zone where $A_i \leq A/N$. For some any constant lower bound on A_i, a large enough N will force a contradiction.

We would like to place *some* constraint on A_i. A straightforward option is to assume the maximum to minimum zone area is bounded:

$$\frac{A_i}{A_j} < K$$

for all i and j, and for some constant K. This implies that for all i:

$$\frac{A}{KN} < A_i < \frac{KA}{N}$$

We would also like to say something about p_i. In general, there is no reason that all rays cannot go to the same point with probability one, but this is not physically reasonable. Instead we can assume that the probability of any *particular* point being hit is zero, which is equivalent to

assuming there is a maximum radiance value, L_{max} in the scene. This allows us to establish an upper bound for p_i. Since all radiances are at most L_{max}.

$$E(L_i) = \frac{R_i}{\pi A_i} p_i \Phi \leq L_{max}$$

and thus:

$$p_i \leq \frac{\pi A_i}{R_i \Phi} L_{max}$$

So our bound for variance of L_i becomes:

$$var(L_i) \leq \frac{R_i}{\pi A_i} L_{max} \frac{\Phi^2}{r}$$

For physical reasons, we can also assume all reflectances are bounded:

$$0 < R_i \leq R_{max} < 1$$

From previous bounds on A_i (all areas at least $A/(KN)$), and R_i we have for all i:

$$var(L_i) \leq \frac{N R_{max}}{\pi K A} L_{max} \frac{\Phi^2}{r}$$

Grouping the constants into a new constant C yields:

$$var(L_i) \leq C \frac{N}{r}$$

To ensure that we estimate all L_i with a variance lower than V_0, we can simply use:

$$r = \frac{CN}{V_0} \tag{3}$$

Thus, the number of rays needed for this Monte Carlo simulation is $O(N)$, where N is the number of zones.

3 Including Indirect Lighting

One way to extend the Monte Carlo method is to use a progressive refinement technique [3, 1, 14], where each zone, in turn, sends the power it received in direct lighting. This process could continue until most of the power is absorbed. Unfortunately, it is algebraicly difficult to extend the reasoning from the last section to such this algorithm.

Instead, we can consider a simpler (probably less efficient) Monte Carlo strategy. As before, we emit r rays each carrying Φ/r power. When a ray hits a zone z_i, it is reflected with probability R_i and absorbed with probability $(1 - R_i)$. In other words, they behave like photons. Each zone will reflect some number of rays, r_i, and absorb some number, a_i. Note that the sum of all a_i will be r, while the sum of all r_i is unbounded (the expected value of that sum is bounded). We have three possible unbiased estimators for Φ_i. We could simply count the total power of reflected rays:

$$\Phi_i \approx r_i \frac{\Phi}{r}$$

Or we could use the reflectance times the power hitting z_i:

$$\Phi_i \approx R_i(r_i + a_i)\frac{\Phi}{r}$$

Or we could use the rather indirect estimator:

$$\Phi_i \approx \frac{1 - R_i}{R_i} a_i \frac{\Phi}{r}$$

The easiest estimator to use, for the purposes of our proof, is the third. This is because it takes the form of a simple sum of r elements: each of the r rays is absorbed by exactly one zone. The first two estimators are probably lower variance estimators, but both can allow a ray to interact with more than one zone, so any analysis using those estimators would be more involved than analysis using the third estimator. This allows us to bound the variance of L_i:

$$var(L_i) \leq \frac{N R_{max}}{(1 - R_{max})\pi K A} L_{max} \frac{\Phi^2}{r}$$

so as before, the to get below our variance threshold, we need $r = O(N)$.

Since each original ray from the light source may reflect off many surfaces, the expected number of *total* rays, the number of intersection calculations needed, as opposed to the number of original power carrying paths, is r times one plus the average number of reflections for each packet. Because reflectivity is bounded, we have:

$$E(r_{total}) \leq r(1 + R_{max} + R_{max}^2 + R_{max}^3 + \cdots) = \frac{r}{1 - R_{max}}$$

So, choosing r according to (3) ensures that we will satisfy our target variance condition and:

$$E(r_{total}) \leq \frac{CN}{V_0(1 - R_{max})} = O(N)$$

Thus, the expected number of rays traced is $O(N)$.

4 Experimental Tests

Figure 1 shows twelve different images calculated by progressive refinement zonal ray tracing. As in the simple simulation described in the last section, each ray carries approximately the same amount of power (in other words, a dim patch sends fewer rays than a dark patch). The top row of four images contains 256 zones, the middle row contains 1024 zones, and the bottom row contains 4096 zones. The left hand column is a solution with 4 rays per zone, the second from the left is 16 rays per zone, the second from the right is 64 per zone, and the right column is 256 per zone. In other words, going down a row quadruples the number of zones and quadruples the total number of rays sent, and going right one column quadruples the number of rays sent. If the number of rays needed is indeed $O(N)$, then we would expect the images in each column of Figure 1. This seems to be the case.

A numerical test was devised for a cubical enclosure with constant emitted radiance ($L_e = 0.5$) and reflectivity ($R_i = 0.5$). This has an analytical solution of $L_i = 1.0$. It was tested for $N =$ 1536, 6144, 24576, 98304, 393216, and 1572864. The computational variance of the estimate remained within 2% of constant for $r = 4N, 16N, 64N, 256N$.

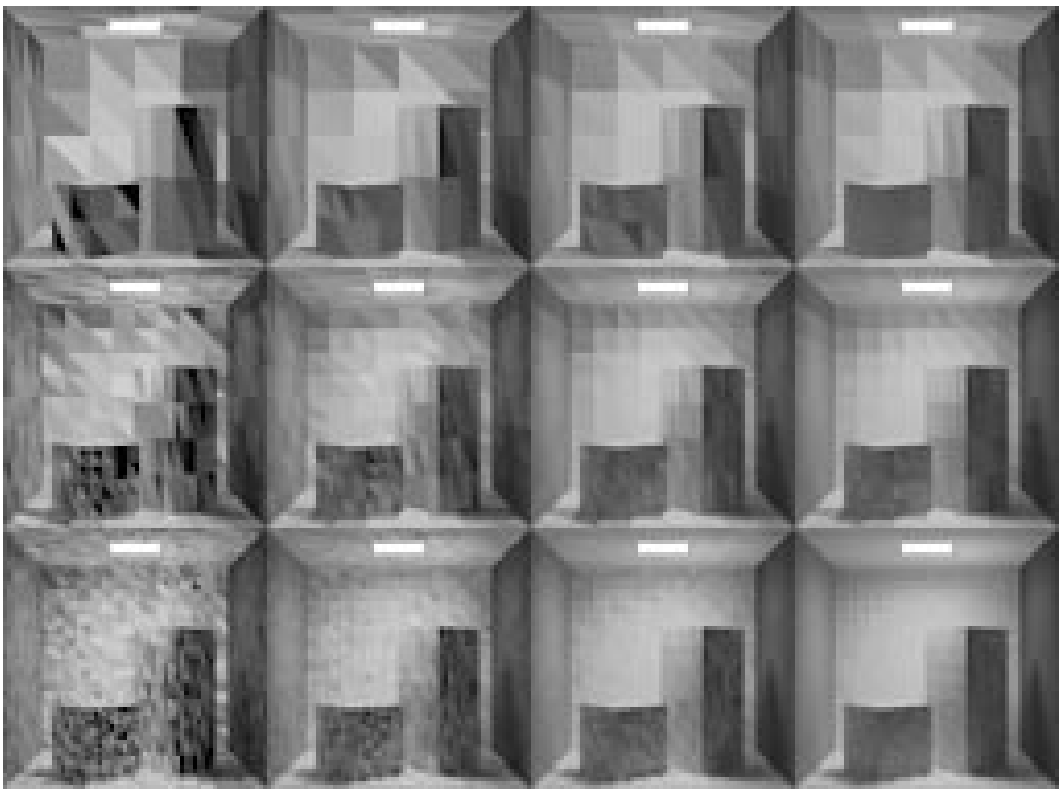

Figure 1: Rooms where each column has a number of rays proportional to the number of zones.

5 Conclusion

The proof that $O(N)$ rays are required for N zone Monte Carlo radiosity relied on three assumptions. The first two, that the maximum radiance in the scene is bounded, and the maximum reflectance is less than one, are physically motivated. The third, that the ratio of largest to smallest zone area is bounded, can be violated in a program, which would mean the $O(N)$ result might not hold.

In practice, the proof does mean that evenly cutting every zone in half should only double the number of rays required, because the area ratios would not change. Intuitively, the area constraint implies that we must send enough rays to accurately find the radiance of the smallest patch.

We still do not have a solution for the time needed, as opposed to the number of rays needed. Ideally, if a divide and conquer search structure, such as an octree [5], is used, the average time to trace a ray can be $O(\log N)$. In practice, this seems to usually be true. This would make our solution time $O(N \log N)$. This could only be proven if we place some restrictions on the geometry of the zones. It will probably be fairly difficult to prove, but such a result of value in the analysis of many graphics problems.

An important practical matter implied $O(N \log N)$ time complexity is that a progressive refinement radiosity program should not search for the brightest patch to shoot power. Such a search is $O(N)$, so the complexity would become (O^2) at best. Instead, the patches could shoot

in random order, or a $O(N \log N)$ sort could be performed once or several times during the execution of the program.

The result in this paper implies that Monte Carlo radiosity may be competitive with other radiosity methods, especially for large N. More investigation into the specific value of the ray constant C is needed to address this issue precisely.

6 Acknowledgements

Thanks to John Airey, Randall Bramley, Jean Buckley, David Salzmann, Changyaw Wang, William Brown, John Wallace, and Kelvin Sung for their help and advice. Special Thanks to William Kubitz, who was the advisor for this project.

References

[1] John M. Airey and Ming Ouh-young. Two adaptive techniques let progressive radiosity outperform the traditional radiosity algorithm. Technical Report TR89-20, University of North Carolina at Chapel Hill, August 1989.

[2] John M. Airey, John H. Rohlf, and Frederick P. Brooks. Towards image realism with interactive update rates in complex virtual building environments. *Computer Graphics*, 24(1):41–50, 1990. ACM Workshop on Interactive Graphics Proceedings.

[3] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics*, 22(4):75–84, August 1988. ACM Siggraph '88 Conference Proceedings.

[4] Micheal F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An efficient radioisty approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(2):26–35, 1986.

[5] Andrew S. Glassner. Space subdivision for fast ray tracing. *IEEE Computer Graphics and Applications*, 4(10):15–22, 1984.

[6] Cindy M. Goral, Kenneth E. Torrance, and Donald P. Greenberg. Modeling the interaction of light between diffuse surfaces. *Computer Graphics*, 18(4):213–222, July 1984. ACM Siggraph '84 Conference Proceedings.

[7] Roy Hall. *Illumination and Color in Computer Generated Imagery*. Springer-Verlag, New York, N.Y., 1988.

[8] Pat Hanrahan and David Salzman. A rapid hierarchical radiosity algorithm for unoccluded environments. In *Proceedings of the Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 151–171, June 1990.

[9] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, July 1991. ACM Siggraph '91 Conference Proceedings.

[10] Paul S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. *Computer Graphics*, 24(3):145–154, August 1990. ACM Siggraph '90 Conference Proceedings.

[11] Malvin H. Kalos and Paula A. Whitlock. *Monte Carlo Methods.* John Wiley and Sons, New York, N.Y., 1986.

[12] Thomas J. V. Malley. A shading method for computer generated images. Master's thesis, University of Utah, June 1988.

[13] Peter Shirley. *Physically Based Lighting Calculations for Computer Graphics.* PhD thesis, University of Illinois at Urbana-Champaign, November 1990.

[14] Peter Shirley. A ray tracing algorithm for global illumination. *Graphics Interface '90*, May 1990.