A Global Algorithm for Linear Radiosity

Mateu Sbert
Xavier Pueyo

Report LSI-93-37-R

A global algorithm for linear radiosity

Mateu Sbert, Xavier Pueyo
Departament d'Informàtica i Matemàtica Aplicada
Escola Politècnica Superior de Girona - Universitat de Girona
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Abstract

A linear algorithm for radiosity is presented, linear both in time and storage. The new algorithm is based in previous work by the authors and in the well known algorithms for progressive radiosity and Monte Carlo particle transport.

Introduction

We will present here a new algorithm for radiosity that is linear, both in time and storage, in the number of patches. This algorithm is based in our previous work [Sbert93], and in ideas borrowed from *progressive radiosity* [Cohen88] and in *particle transport* from the literature about radiative heat transfer [Siegel92] and Monte Carlo radiosity [Pattanaik92, Shirley91].

In progressive radiosity we choose a patch with accumulated unsent energy, on the fly we compute the form factors from that patch to all the others and so we can distribute the energy emitted by it. A patch that so receives energy stores it in two places: an accumulator that at the end of the process will account for the radiosity of the patch, and another accumulator for the unsent energy. That unsent energy is of course updated to 0 whenever the patch is chosen as the shooting patch. In particle transport techniques, first used in Radiative Heat Transfer and introduced in the Radiosity field by Shirley and Pattanaik, we send rays from the sources, and we follow them through its interaction with the surfaces in the environment. Every ray from a source transports an equal quantity of energy, namely the total sent by the source divided by the number of rays we are going to cast from that source. [Sbert93] cast rays in a global way to obtain a form-factor matrix with time cost linear in the number of patches, for a fixed error. Every ray intersected the whole scene and gave as a result an ordered intersection list. That list of intersected patches was followed a pair of patches at a time, updating the number of lines between them and the total of lines crossing each of them. The quotient of the two quantities gave us the form-factor.

We present here two versions of a new algorithm. The first version is a pure global one, not taking into account potentially big local errors. Those errors appear mainly due to the fact that the lines crossing a patch are proportional to the surface of a patch, and then a little source patch will cause a big amount of error. That problem is adressed by the second version of the algorithm.

Overview of the algorithm

From progressive radiosity we borrow the idea of having two quantities per patch, the accumulated energy and the unsent energy. From particle transfer we borrow the idea of rays transporting energy in quanta (discreet amount of energy) from sources to the environment. From our previous work we take the idea of casting global lines (instead of casting them locally at each light source) and building ordered intersections lists. But in our new algorithm, on the contrary to progressive radiosity, we have to wait until the end of the process to have a workable image; i.e. we don't refine it progressively.

We are going to cast globally a determined number of oriented lines. We consider the lines oriented. Each line will supply us with an intersection list, and we follow this list taking into account successive pairs of patches. Each patch (if not emissor) has with it two quantities. One records the energy accumulated, the other one is the unsent energy. When we take one pair of patches along the intersection list , the first patch in the pair will transmit its unsent energy to the second in the pair. So we update to 0 the unsent energy of the first patch, and increment the two quantities at the second patch, the accumulated and the unsent energy. In the case of a source we keep also a third quantity, the emitted energy per line exiting the source. We compute that energy in the following way: Given the number of lines we are going to cast, for any light source we compute beforehand the forecasted lines passing through it. We divide the total energy by this number, and we will have the energy that should transport a single line exiting the surface. Then, if the first patch of a pair is a source patch, the energy transported to the second patch of the pair will be incremented with the emitted energy per line.

The algorithm is then as follows:

```
Initialise emitted energy, luminance and unshot energy for every patch /*Precompute emitted */
                                                  /*energy per ray crossing a source, */
                                                  /*and put to zero every other thing*/
While not all rays cast do /*One pass for every global line cast*/
    Begin
    Cast a ray
    Build sorted list of intersections /*Build an ordered list of patches intersected*/
    For every pair of patches i,j in the intersections list do /*One pair of patches at a time*/
        Begin
        Transfer to patch j the unshot energy of patch i /*As in progressive radiosity*/
        If patch i is emissor do
            Transfer to patch j the energy per ray exiting patch i /*As in particle */
                                                  /*transfer. That energy */
                                                  /*was precomputed */
                                                  /*in the initialisation*/
        Put to 0 the unshot energy of patch i /*As in progressive radiosity*/
        Update accumulated and unshot energy for patch j /*As in progressive radiosity*/
        End for
End while
```

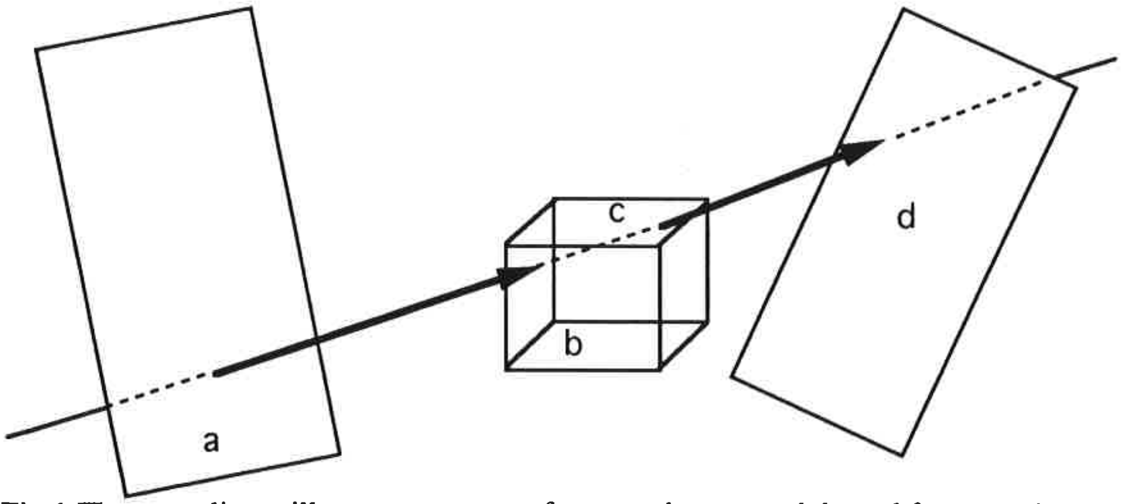


Fig.1 The same line will transport energy from patch a to patch b, and from patch c to patch d.

Complexity

We will show that the error of the new algorithm decreases linearly with the number of lines and increases linearly with the number of patches. Suppose Li* is the exact solution for patch i, Li is the solution with the new algorithm and Li' is the solution with the algorithm presented in [Sbert93].

For any Li, Li*, Li' values we have

$$(Li^*-Li)^2 <= 2\ (Li^*-Li')^2 + 2\ (Li'-Li)^2 \qquad (1)$$

This fact is easily proven. Developing the squares, and passing all terms to one side of the inequality, we would obtain

$$(Li^*+Li-2Li')^2 >= 0$$

Then, summing (1) for all the patches

$$\Sigma_i\ Ai\ (Li^*-Li)^2 <= 2\ \Sigma_i\ Ai\ (Li^*-Li')^2 + 2\ \Sigma_i\ Ai\ (Li'-Li)^2$$

As shown in [Sbert93] the first term after the sign of inequality is linear in the number of patches. If we now show that the second term is linear too, we should have proved our assertion. We have verified the linearity of the second term empirically. We have performed some tests based in different scenes and different number of patches, all based in 100 runs. Those tests, summarised in Tables 1&2, appear to confirm the desired linearity. The error in those tables is the square of the RMS error, that we term as quadratic error. Finally, in Table 3 we have computed the error respective to a solution computed with exact (analytical) form factors. Again, there seems to be linearity.

As in our new method we do not compute neither explicitly nor implicitly the form-factor matrix, but instead compute directly the luminance vector, it is clear that our cost in storage is linear in the number of patches.

	number of patches		
lines	6	54	600
10^3	0.0009254	0.0073325	-
10^4	0.0000885	0.0008175	0.0079969
10^5	0.0000081	0.0000796	0.0008868
10^6	-	0.0000080	0.0000888
10^7	-	-	0.0000088

Table1. Quadratic error (square of RMS) for scene 1, a cubical enclosure. The error is computed respective to the solution found with the method of [Sbert93].

lines	number of patches 114	456	1824
10^4	0.0010487	0.0041131	-
10^5	0.0001092	0.0004458	0.0017342
10^6	0.0000113	0.0000453	0.0001792
10^7	-	-	0.0000180

Table2. Quadratic error (square of RMS) for scene 2, a cubical enclosure with three cubes inside. The error is computed respective to the solution found with the method of [Sbert93].

lines	number of patches 6	54
10^3	0.0007985	0.009267
10^4	0.0000789	0.0007775
10^5	0.0000079	0.0000801
10^6	-	0.0000080

Table3. Quadratic error (square of RMS) for scene 1, a cubical enclosure, respective to the analytical solution.

Improving the algorithm.

We present here two ways of improving the performance of the algorithm. First we propose an hybrid algorithm, consisting of one or more local passes, and finally the global pass. Second, when considering a given pair of patches, we exchange the unsent energy in both directions.

We are burdened with two facts: *First*, the work done with the first shot lines may be wasted in the sense that probably they do not transport energy at all. That is, much work is done to put the system in a state where each line transports energy between every pair of patches. Only then, when most of the patches have unshot energy the system will work at its best. *Second*, as it is a global algorithm, the lines passing through any source will be proportional to the surface of the source, not to its strength, and then the error may be a big one for small area sources. In fact, the standard deviation of λ, the expected number of lines through a patch, will be $\sqrt{\lambda}$ (considering Poisson approximation to binomial distribution), so for instance for an expected value of 100 lines the standard deviation is 10, that is a 10%. It would mean that we expect to make an error of about a 10% in the total energy sent from that source, so a big error. If λ is 10000, the relative error comes down to 1%, a more acceptable error. But if we have to make sure that for each small source pass a minimum of 10000 lines, the total quantity of lines cast would be enormous, and our method would become not very practical.

Then the question is: May we put the system in a working state in a direct and local way?. As often done, the answer is as follows:

1-Shoot energy directly from all the sources, in a local Monte Carlo way, or in any other way.

2-Put to 0 the energy of the sources, and initialise the energy of the patches to the irradiated (received) energy, factored by the reflectivity.

3-Follow the algorithm proposed in the previous section with no additional change.

4-Sum to the final result the initial energies.

We see that we do not add complexity, because the work done in the first shot is O(n), where n is the number of patches.

The algorithm could be again modified in the sense that before applying part 2, we could shoot again from the patch or patches with most unshot energy, like it is done in progressive radiosity. Part 2 to 4 could account then for the slow convergence of progressive radiosity, and we should have then an hybrid algorithm with the advantages of both progressive radiosity and global radiosity.

Some tests are summarised in Tables 4&5. Table 4 shows the quadratic errors (square of RMS error) when the first shot is an exact one. That is, we have computed analytically the form-factors from the source to the other patches with the formulae in [Siegel92] and distributed the energy according to those form-factors. Compared to Table 3, Table 4 shows a decrease of an order of magnitude. If we look at Table 5, where we have put the respective quadratic errors for the two versions of the algorithm, and we have supposed a much more smaller light source than in Table 4, we see that the decrease is about 50 times. The smaller is the light source and the more strength it has, the more will the decrease be.

	number of patches	
lines	6	54
10^3	0.0000833	0.0009188
10^4	0.0000083	0.0001084
10^5	0.0000008	0.0000112
10^6	-	0.0000011

Table4. Quadratic error (square of RMS) for scene 1, a cubical enclosure, respective to the analytical solution, with the first shot computed with analytical form factors. Compare with Table 3.

lines	first version	first shot exact
10^3	0.0457937	0.0008846
10^4	0.0051473	0.0001051
10^5	0.0005056	0.0000105
10^6	0.0000505	0.0000011

Table5. Quadratic error (square of RMS) respective to the analytical solution for scene 3, a cubical enclosure divided in 54 patches and a light in a patch in the middle of a face, with as much energy as scene 1.

As a second way to improve the algorithm we propose that, when considering a given pair of patches, we exchange the unsent energy in both directions. That is, if we do not take into account the orientation of the intersections list, then we have that every line joining two patches represents a two-directional path between them. That is, each of them sends simultaneously to the other the unsent energy. In this way we may spare a 50% of lines cast. Table 6 gives some error results. This last improvement is used to compute the radiosity of the images presented here. Figures 3&4 show the distribution of the quadratic error for the runs corresponding to Table 6. The quotient of the standard deviation by the mean (Coefficient of Variation) is for 6 patches 77.1% for the first case and 80.6% for the second case, and for 54 patches 20.4% and 22.3% , respectively. The graph shapes for both 6 and 54 patches look like those of χ^2 distributions, with a few degrees of freedom for 6 patches and about fifty degrees for 54 patches. This is also consistent with the values of the Coefficient of Variation for those χ^2 distributions. If this ressemblance to a χ^2 distribution is due to the simplicity of the scenes considered, or is valid for most scenes, must be further investigated. As the Coefficient of Variation becomes smaller incrementing the degrees of freedom, a χ^2 behaviour would mean increased stability with increased number of patches, that is, less fluctuation of the error.

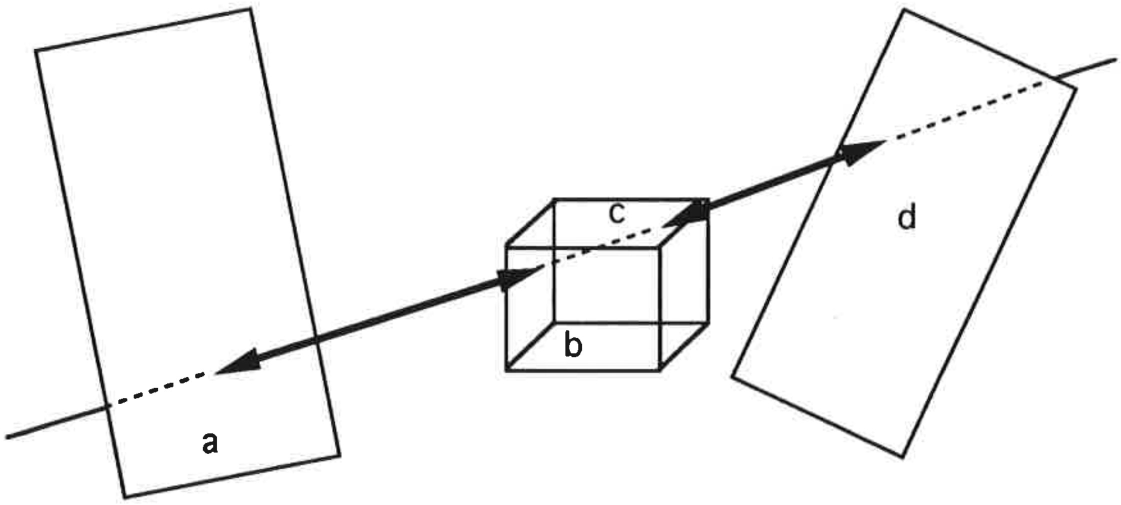


Fig.2 The same line will transport the unsent energy from patch a to patch b, and simultaneously the unsent energy from patch b to a. The same with patches c and d.

	number of patches	
lines	6	54
10^3	0.0003190	0.0031117
10^4	0.0000298	0.0003121
10^5	0.0000039	0.0000315
10^6	-	0.0000031

Table6. Quadratic error (square of RMS) for scene 1, a cubical enclosure, respective to the analytical solution, taking into account bidirectionality of a line between two patches.

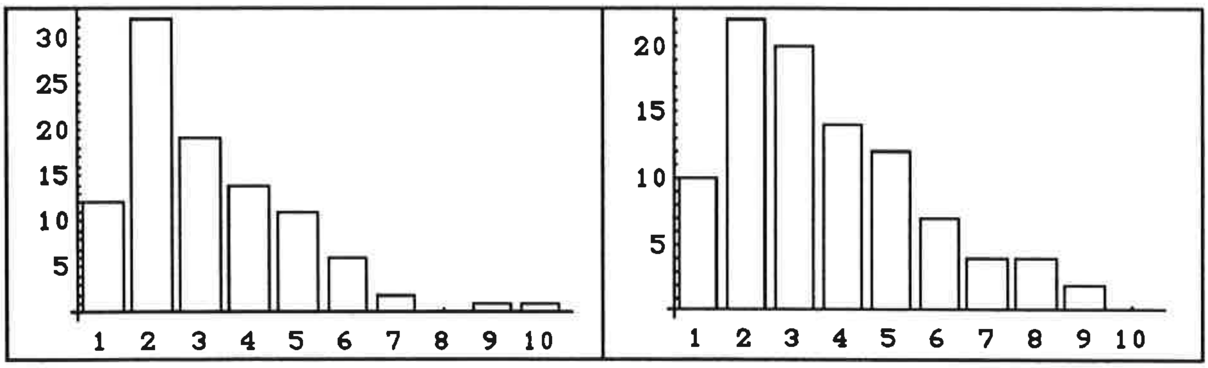


Figure 3. Graphs of frequencies of the errors of the 100 runs corresponding to the 6 patches case in Table 6, for 10000 lines and 100000 lines, respectively. Errors are given in 10^{-5} units in the first case, and in 10^{-6} units in the second case.

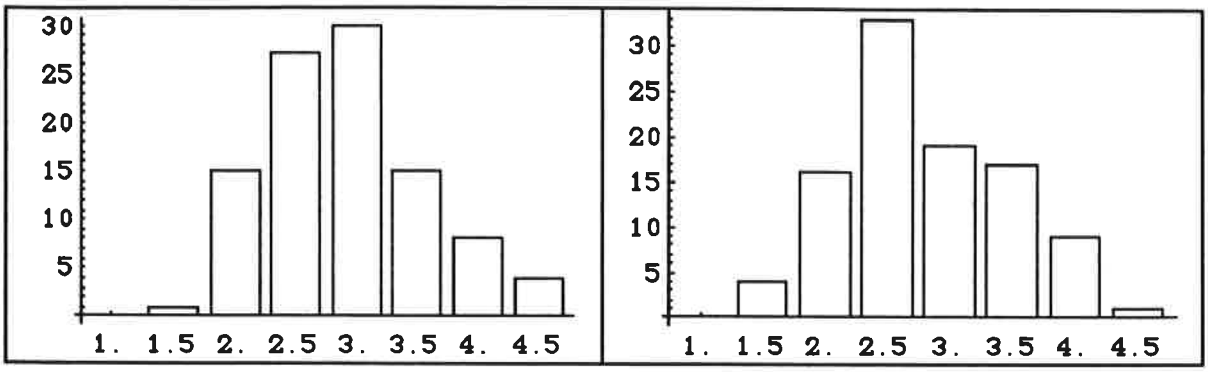


Figure 4. Graphs of frequencies of the errors of the 100 runs corresponding to the 54 patches case in Table 6, for 100000 lines and one million lines, respectively. Errors are given in 10^{-5} units in the first case, and in 10^{-6} units in the second case.

We present in Figures 5&6 two images obtained making use of the algorithm with both improvements presented in this section. The scene consists of a cubical enclosure with three cubes inside. The average length of an intersection list is of 1.38, and of a non-empty intersection list of 2.33. The number of patches is 12150. The source is the whole front wall (not seen in the image). In Figure 5 we have cast eight millions of local rays from the source, and twelve millions of global rays. The average number of lines per patch of the enclosure is 1474, per patch of the big cube is 1164, and for the smaller cubes 655. In Figure 6 we have cast sixteen millions of local rays from the source, and twenty four millions of global rays. The average number of lines per patch of the enclosure is 2948, per patch of the big cube is 2328, and for the smaller cubes 1310.

Conclusions and future work

We have presented here a new Monte Carlo algorithm for computing the radiosity of a scene, based on three existing techniques: progressive radiosity, particle transport, and global Monte Carlo. The cost have been shown to be linear in time and space with respect to the number of patches, for a given scene and a global error bound. We have also presented two improvements to the basic algorithm, based on considering a first local shot, and then switching to the global method, and also in considering transfer of energy along two directions for a same line. We have shown how at least for very simple scenes, the method is stable, in the sense that the fluctuations of the error decrease with increased number of patches. We have also shown how for those very simple scenes the error distribution looks like a χ^2 distribution. Nevertheless, much work must be done along the following lines:

-analysis of the error

A thoroughly analysis of the error must be done, focused in two points. The first point will be to see if the error follows a χ^2 distribution and which degrees of freedom it has. The second point is to find an heuristic formula which gives us an a priori value of the error, based on the parameters of the scene. This formula will permit us to compute the number of lines necessary for a given scene and a given error.

- progressive radiosity

A natural continuation of the work presented in this paper, as well as the one presented in [Sbert93], is to find a progressive refinement approach well suited to the global Monte Carlo integration. In [Sbert93b] we propose a solution, wich could be extended to the technique presented here.

-non-planar surfaces

In [Sbert93b] we show how we may use scenes with non-planar parametric surfaces. A future work would be to show how the presented algorithm is also well suited to those same scenes.

- non-diffuse surfaces

We are going to extend the algorithm in the following way, not yet implemented: If the intersected surface is a non diffuse one, we record the intersection point and the direction of intersection. When we follow the intersection list from a patch onto the next in the list, if we detect a non-diffuse surface as the patch receiving the energy, we trigger a procedure to recursively transport that energy, as does Shirley [Shirley90]. That procedure finishes either when the light lands on a diffuse surface, or when its energy has diminished below some preestablished threshold.

-bundles of parallel lines

In [Sbert93b] we show how we may use random bundles of parallel lines to compute form factors, the main advantage of this being that we could project the whole scene in the direction of the bundle and then use incremental methods, as described by Bucklaew [Bucklaew89]. A future work would be to show how those bundles may also substitute our random lines in the algorithm presented here.

Acknowledgements

The authors wish to thank Pere Brunet, Josep Vilaplana, Joaquim Gelabertó, Carles Barceló and Joan Surrell for their helpful comments, and Frederic Pérez and Albert Vergés for helping with the software. This project has been funded in part with grant number TIC 92-1031-C02-01 of the Spanish Governement, and a grant PIAR'92 from the Generalitat of Catalonia .

References

[Buckalew89] BUCKALEW, C. , FUSSELL, D. , "Illumination Networks: Fast Realistic Rendering with General Reflectance Functions." *ACM Computer Graphics* , Vol. 23, Number 3 July 1989.

[Cohen88] COHEN,M.F., CHEN,S.E., WALLACE, J.R., GREENBERG, D.P. "A Progressive Refinement Approach to Fast Radiosity Image Generation", Computer Graphics, Volume 22, Number 4, August 1988 (SIGGRAPH'88)

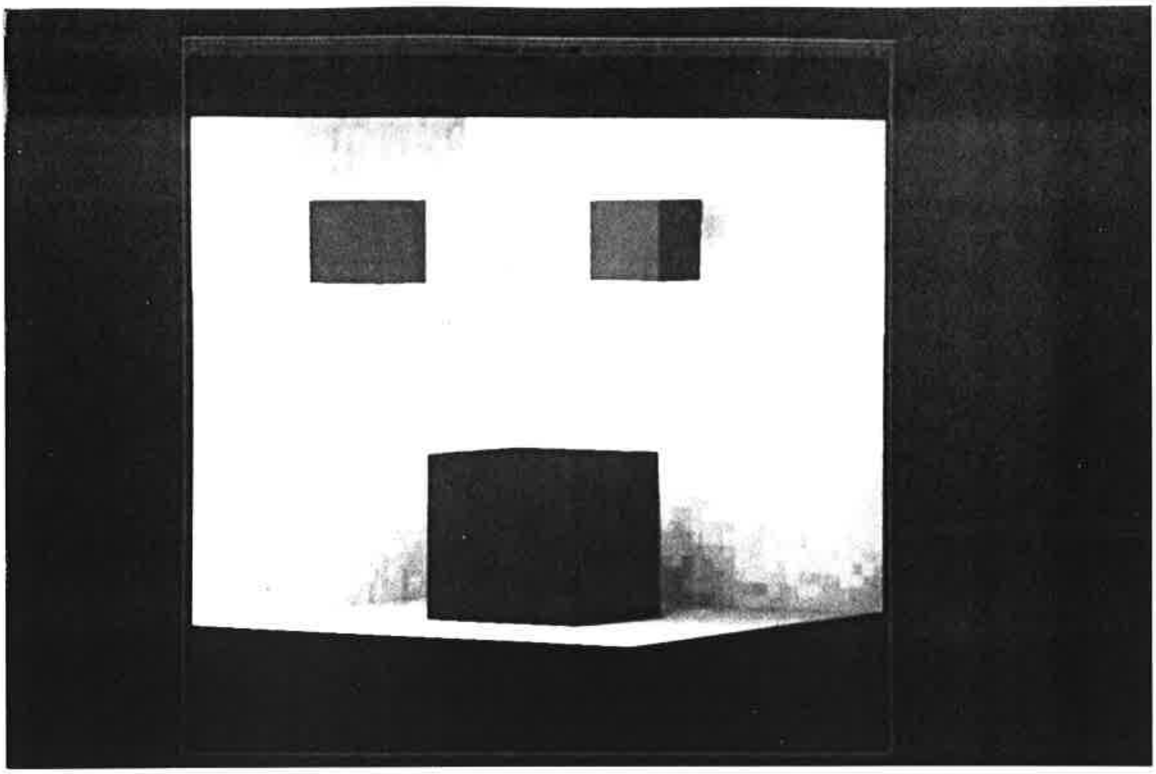
[Pattanaik92] PATTANAIK,S.N., MUDUR, S.P. "Computation of global illumination by Monte Carlo simulation of the particle model of light" Third Eurographics Workshop on Rendering Conf. Proc. (May 1992), pp.71-83.

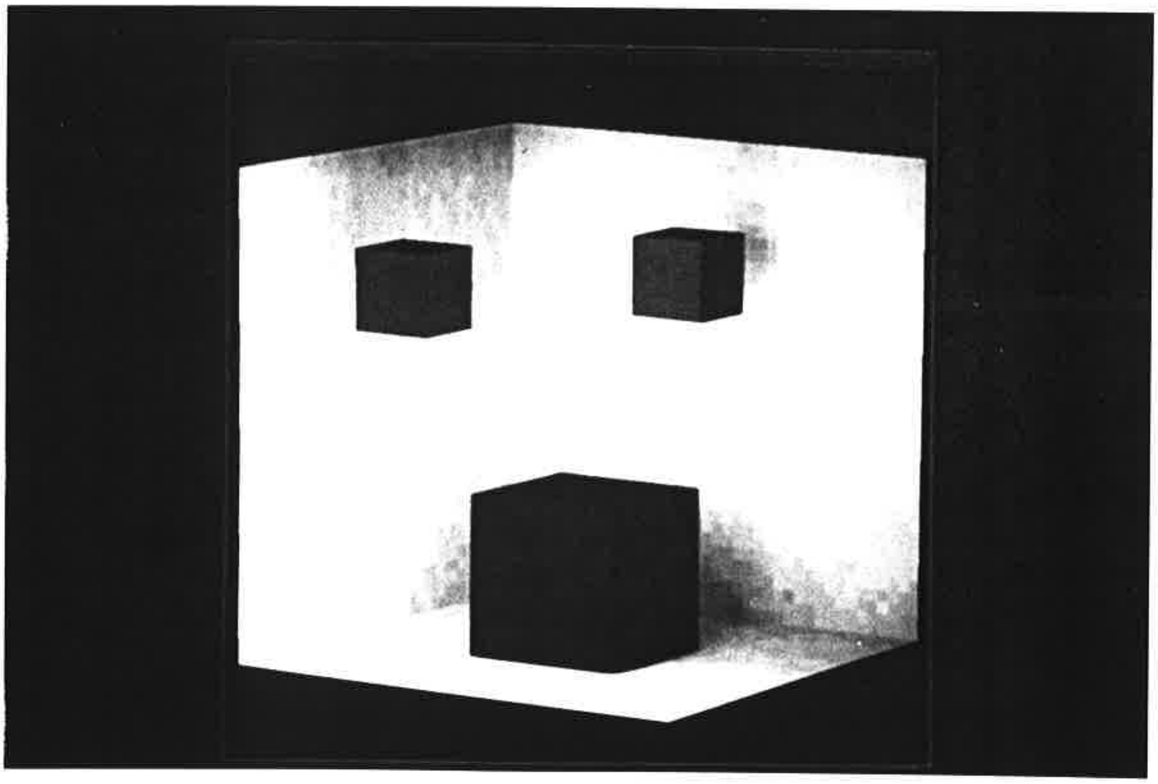
[Sbert93] SBERT, M., "An Integral Geometry Based Method for Fast Form-Factor Computation." Computer Graphics forum, vol.12, n.3, pp.409-420 (proc. Eurographics'93)

[Sbert93b] SBERT,M., PUEYO,X. "Extending the global integration method." In preparation.

[Shirley90] SHIRLEY,P. "A Ray tracing Method for Illumination Calculation in Diffuse-Specular Scenes", Graphics Interface'90, 205-211.

[Siegel92] SIEGEL, R., HOWELL, J.R. *Thermal Radiative Heat Transfer* , Hemisphere Publishing Corporation, Washington D.C., 1992





Figures 5 (top) &6 (bottom): Scene with 12150 patches. 8 and 16 millions of local rays, respectively, and 12 and 24 millions of global rays, respectively.