An Integral Geometry Based Method for Fast Form-Factor Computation

Mateu Sbert

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

Monte Carlo techniques have been widely used in rendering algorithms for local integration. For example, to compute the contribution of a patch to the luminance of another. In the present paper we propose an algorithm based on Integral geometry where Monte Carlo is applied globally. We give some results of the implementation to validate the proposition and we study the error of the technique, as well as its complexity.

1. INTRODUCTION

Monte Carlo methods in radiative heat transfer, and thereafter in radiosity, can be classified in two: those that ignore the form-factor matrix and only need a final solution, and those which explicitly give the form factor matrix and solve *a posteriori* the equations system. [Shamsundar73], [Pattanaik92] and [Shirley90] fall into the first category, while [Weiner65] belongs to the second. The first category is more suitable for a quick solution, while the second allows the possibility of changing the problem's initial conditions of illumination. As [Siegel81] puts it: "...unless a parametric study of the interchange of radiant energy within an enclosure with specified characteristics is being carried out, it may be easier to directly compute the entire radiative flux distribution by Monte Carlo. This would be simpler than computing configurations factors by Monte Carlo and then using an auxiliary program to calculate energy exchange by means of these factors." We could then speak of non-parametric and parametric methods, where the parameters are, of course, which patches are light sources and their initial luminance.

We could draw the same distinction between non-Monte Carlo methods: one type of method makes use of progressive radiosity [Cohen88], and the other ones calculate explicitly the form factor matrix, and then solve the system of equations. Following [Baum89], we have Full-Matrix methods (F-M), and Progressive Radiosity methods (P–R).

The difference in cost between the two types of methods stems from the fact that the calculation of form factor matrix is quadratical in the number of patches (Recently [Hanrahan91] presented an algorithm with time less than quadratical in some circumstances, and worst time quadratical). There is also another difference, that of the storage cost of the form factor matrix.

In fact, the cost in time of progressive radiosity should be the same as the Full-Matrix, the main difference being that we have almost at once an usable image of the scene, while in Full-matrix we have to wait until all the form factors are calculated: we can stop the iterations at some level, without a full convergence, and so balance time with precision.

There is, therefore, a dilemma: speed versus reusability. This paper addresses this dilemma and presents a Monte Carlo method based on Integral Geometry for finding form factors lineal in time with respect to the number of patches. The proposed method takes an approach that could be termed as global integration, versus local integration, that is one patch at a time, of previous methods.

In the next section we present an overview of the proposed algorithm. In section two we present three possible estimators for the form factor. Section three discusses the errors introduced in the computation of both, form factors and visibility. In section four we study the complexity of the algorithm. Finally, we present the results in section five as well as our conclusions and proposed future work in section six.

2. PROPOSED METHOD

Integral Geometry [Rey-Pastor51, Santaló76] allows us to study and measure sets of lines, for example how many lines intersect a convex body, how many intersect a surface, what proportion of lines intersecting a convex body will also intersect a second body contained in the first. Radiosity [Goral84, Greenberg86] studies the transfer of energy between the diffuse surfaces of a scene, and its equilibrium. Energy is transfered through light rays, and obeys Lambert's cosine law [Meshkov81].

On the other hand, lines are passing through a surface in every direction, and Integral Geometry teaches us that the flux or line density going off a differential area $d\sigma$ in direction θ is given by $cos\,\theta\,d\sigma\,d\omega$, the same law as Lambert's.

If we integrate over the whole surface S (supposing it is a planar one), we get as the measure of the total number of lines intersecting it πA , where A is the area of S. In consequence, the fraction of this total going from S to surface S' will be the quotient

$$\int cos\theta\,d\sigma\,d\omega\,/\,\pi A$$

where the integral extends for every $d\sigma$ to the solid angle projected by S', and $d\sigma$ goes over the whole surface S. This formula, as it should be, is the same found in Radiosity with the name Form Factor between surface S and S' ($d\omega$ is also found as $cos\theta'\,d\sigma'/\,r^2$, where $d\sigma'$ is an area differential in surface S', θ' is the angle formed by segment r joining both area differentials with the normal to $d\sigma'$).

Making use of the Laplace Rule then we interpret the form factor as a probability , and use indistintly the methods of Integral Geometry or its related topics, geometric probabilities. We take a random sample of the lines intersecting a surface, see what other surfaces are first intersected, and by simply counting the intersections for every surface and dividing by the total we estimate the form factors of the first surface respective to the others.

2.1 Overview of the algorithm

A suggestive idea to make the most of ray casting is as follows: Inscribe the scene in a sphere, and take on the sphere two random points. Intersect the scene with the line through those two points. The surfaces will be intersected in pairs, one for every body intersected. Sort the intersections by distance from one of the points (see Figure 1). If we cast r random lines, at the end we'll have r sorted lists of intersections from which we may derive n visibility lists where n is the number of patches of the scene. These visibility lists were previously used in [Buckalew89], but the rays were not cast randomly. In the present method, the rays cast in a random manner are used not only for visibility purposes but also to compute form factors. Each line drawn may contribute to the computation of several form factors. So, instead of locally tracing lines for each pair of patches, we trace lines for the whole environment, each shot contributing to several patches.
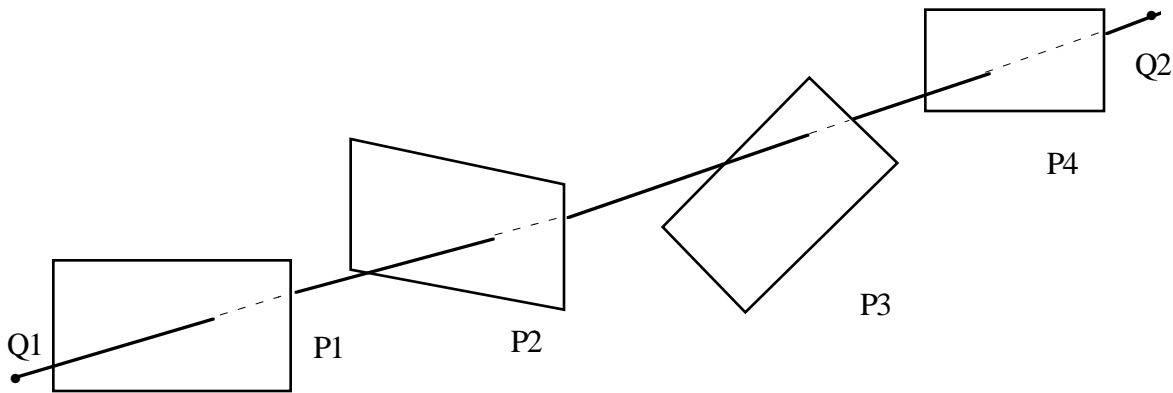

Fig.1 If we order according to the distance from Q1, we get list (P1, P2, P3, P4) , from which we conclude P1 and P2 both see one another, and the same with P3 and P4, the path between P2 and P3 being in the interior of some body bounded by (among other patches) P2 and P3. So we add P2 to the list attached to P1, P1 to the list attached to P2, P3 to the list attached to P4, and P4 to the list attached to P3.

We could summarize the method in the following algorithm:

```
Initialize visibility lists
Repeat until no more rays to cast
        Begin
                Put intersections list to empty
                Cast a random ray
                For every surface in scene do
                        Begin
                                Look for intersection
                                        If intersected do
                                                Find patch
                                                Add to list
                                        End if
                        End;
                Sort list
                Update visibility lists
                Update number of intersections of each patch
        End;
```

A visibility list records how many times a pair of patches see one each other. So, how many lines intersect both.

2.2 Estimation of form factors.

The average of the number of lines intersecting a patch with area A is *Nr* times *$2A/4\pi R^2$*, where R is the radius of the surrounding sphere and Nr is the number of rays cast. This formula comes from Integral Geometry, which states that the measure of a set of lines intersecting a convex body is π *Area of the body / 2* . In the case of a sphere, *$4\pi^2R^2/2$* . We have seen above that the measure of lines intersecting a planar surface is πA (we could also consider a prism with base our patch and infinitesimal height ε , and so the measure of the set of lines crossing the body will be $\pi(2A+\varepsilon)/2$, or πA in the limit) . Then, dividing, we get $2A/4\pi R^2$ as the probability that a line that crosses the sphere also intersects the patch. If we now multiply by the total number of lines, we have our expected value. The number of lines intersecting a patch follows a binomial distribution, with parameters *Nr* and *$2A/4\pi R^2$* , that we may approximate by a normal distribution. If, for example, the average for a patch were 100 lines then the probability of, suppose, less than 70 lines intersecting the patch is about 0.001. We cannot conclude that 1/1000 of the patches will have received less than 70 lines, because the distributions of a patch with respect to any other one are not independent, but we believe that there is little deviation. So the cost of drawing more intersecting lines for those patches is minimal, but we must provide a method for doing so, especially if those patches are emissors or highly reflectant. What we do is cast more lines, imposing on them the condition of having to intersect the specific patch, their number depending upon some preestablished threshold. The same would be true for those very small patches with a low average number of lines.

2.3. Three form factors estimators

We have at hand three estimators for the form factor between surface i and j, F_{ij}. The first one, and most obvious, is to divide the number of lines crossing the two surfaces at once, r_{ij}, divided by the total of lines crossing surface i, *n*. That is
$F_{ij} = r_{ij} / n$
The second one is as follows: We determine first the expected number of lines passing through surface i, say λ, and then:
$F_{ij} = r_{ij} / \lambda$
The value for λ is : *m* $2A_i$/ *Ssphere*, where *m* is the total of lines cast, *Ssphere* the surface of the surrounding sphere and A_i the area of surface i.

We still have a third estimator: In the denominator, in place of n, as the first estimator, or λ, as in the second, we place a ponderated value on n and n', the total of lines crossing surface j. Specifically, our new denominator is $(A_j n + A_i n')/2 A_j$. Then:

$$F_{ij} = \frac{2A_j\, r_{ij}}{A_j n + A_i n'}$$

In order to test experimentally the three estimators, we devised a test scene: a cubical enclosure where each face is divided in nine equal patches, that is, 54 total patches. For this scene, we calculated the analytical values for the form factors from [Siegel81] and [Hsu67]. We give the values for the mean square quadratical error with the solutions obtained for the three estimators. The initial conditions were initial luminance 1 for patch 42 (in the middle of an edge), and reflectivities {0.11, 0.12, 0.13, 0.14, 0.15, 0.16} for the six faces. The best estimator was the ponderated one, followed by the first one, and worst was the second one. The mean square quadratical errors are given in table 1.

Estimator 1	Estimator 2	Estimator 3
0,00212457	0,0023735	0,0019243

Table 1. Mean square quadratical errors

We show in figure 4 the luminances over the 54 patches, obtained with the exact or analytical form factors and the three estimators just defined.

3. ERRORS
3.1 Error in form factor

Our theoretical study of error is simply based on the fact that we are estimating a proportion (we use the first estimator defined in subsection 2.3.), and accordingly we use the corresponding formulas for the error, [Freund80, Weiner65]. The error will depend on the number n of lines passing through the patch, and we take as a typical value n=10.000. This is in accordance with the values used in ray casting methods to calculate form factors. Considering the normal distribution approximation to binomial, we have with probability (1-α)

$$\left|\theta\text{-}\theta'\right| \leq \frac{z_{\frac{\alpha}{2}}\sqrt{\theta(1-\theta)}}{\sqrt{n}}$$

θ being the proportion to be estimated (in our case, the form factor to be estimated), and θ' the estimated value. A bound for relative error $|\theta\text{-}\theta'|/\theta$ with probability (1-α) will then be

$$\frac{\left|\theta - \theta\right|}{\theta} \leq \frac{z_{\frac{\alpha}{2}}\sqrt{\frac{(1-\theta)}{\theta}}}{\sqrt{n}}$$

The value for $z_{\alpha/2}$ is found in the tables of normal distribution. In Monte Carlo methods [Sobol76] α=0.5, and then the value for $z_{\alpha/2}$ is 0,675. The so obtained error bound is the probable error, because the probability of getting an experimental error lower than this bound is equal to the probability of getting an error greater than this bound. Table 2 show the relative errors for distinct values of form factors (or equally, distinct values of θ, i.e. the proportion of lines that going off first surface land in the second), for n=10000, and in table 3 we have the case $\theta = 0.001$ tabulated with Poisson distribution:

θ	z=0,675
0.2	1,35%
0.1	2,03%
0.05	2,94%
0.01	6,72%
0.005	9,52%

Table 2. The first column lists the form factor, the second the probable error for n=10000

θ	probability= 1−α	relative error <=
0.001	P(8<=x<=12)= 0,5946	20%

Table 3. The first column lists the form factor, the second the probability of the relative error being less or equal than the value appearing in third column. That is, the third column is an upper bound for relative error with probability described in the second column. Variable "x" represents the number of lines passing through both patches.

To test the method experimentally, we devised the following test scene: Consider two unit squares, with a common edge. If we vary the angle between them, we can compare our solution with the analytical one [Spencer90].

We cast 100.000 rays on a sphere centered in (0,0,0) and radius 4. One of the squares is in plane z=1, with vertices (0,0,1), (1,0,1), (0,1,1), (1,1,1), and the second has common vertices (0,0,1) (0,1,1) and inclined an angle β with the first one (see Figure 2).

Table 4 gives the form factors obtained by dividing the lines passing through both squares by the total number of lines passing through the first square. This total is between parenthesis, and from it we have calculated the probable relative error, given in the last column.

Inclination angle	analytic solution	our method	relative error	pr.relative error
30°	0.6190	0,6066 (1009)	2,01%	1,7%
60°	0.3709	0,3872 (997)	4,38%	2,8%
90°	0.2000	0,2163 (957)	8,15%	4,36%
120°	0.0866	0,0904 (1007)	4,35%	6,9%

Table 4. The second column gives the analytical solution, the third the calculated form factor from the first square respective to the second (between parentheses the total lines intersecting the first square), the fourth column the relative error, the fifth the probable error. The expected lines through each square is 995.

3.2 Error in visibility

Let us suppose we want to use our method to determine only the visibility, that is, to build a visibility graph. To know how much we are mistaken, we can find the probability that, if a form factor is non null, we give zero as its value. This probability is $(1-\theta)^n$ because the event is just "line going off the first surface and not intersecting the second one" repeated n times. This value may also be approximated, using Poisson distribution, as $e^{-n\theta}$. Table 5 give the results for 10000 lines.

θ	visibility
0.2	100%
0.1	100%
0.05	100%
0.01	100%
0.005	100%
0.001	100%
0.0001	63,21%

Table 5. In first column we have the form factor, in the second one the visibility, or probability of estimating as non null the form factor, for n=10000

Another question relating to visibility is that the proposed method allows for a patch that partially sees another patch. This partial vision is reflected in the value of the form factor. In Figure 3, the form factor from P1 to P2 is between 0 and the value it should have supposing there was not occlusion by P3 , say θ. That is, moving P3 between P1 and P2 will continuosly vary the form factor from θ to 0 and viceversa.
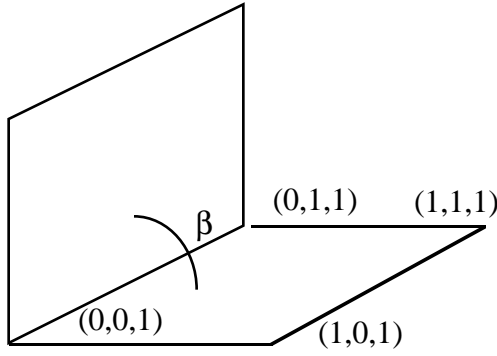
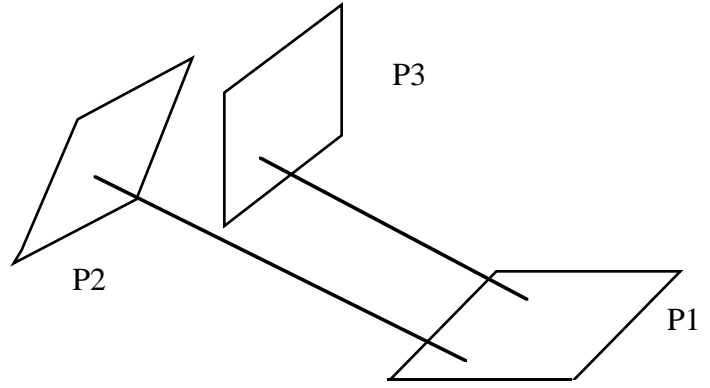Fig.2. Two test squares. β will be given values 30º, 60º, 90º y 120º

Fig.3. P3 partially occludes P2

3.3 Error in luminance

What we are interested in is not the error in form factor, but to what extent this error determines the luminance error. Interestingly, one can give an upper bound for the error in the luminance of any patch that does not depend upon the values of the form factor calculations [Sbert93]. That is, it is a *worst case bound*. Assume that the sum of all calculated factors for a given patch is at most 1, that the same calculated factors fulfill *AiFij=AjFji*   for any two patches i,j , and that all reflectivities are less than 1. Then we can give the bound for the luminance Li of any patch i with reflectivity Ri and initial luminance $L_i{}^0$, where the maximum is taken from all patches.

$$L_i \leq R_i \, max_j \left(\frac{L_j^0}{1\text{-}R_j} \right) + L_i^0$$

For the error in *Li* we have the upper bound:

$$R_i \, max_j \left(\frac{L_j^0}{1\text{-}R_j} \right)$$

But for our Monte Carlo method we can give a much better error estimation, that is, a variance, if we limit ourselves to the study of the error in the first iteration when solving the radiosity n equations systems by an iterative method [Sbert93]. Suppose there are only two emitting patches, r and s. Then the variance is:

$$Var\left(L_i^1\right) = R_i^2 \left(\left(L_r^0\right)^2 \frac{F_{ir}\left(1\text{-}F_{ir}\right)}{n} + \left(L_s^0\right)^2 \frac{F_{is}\left(1\text{-}F_{is}\right)}{n} - 2\, L_r^0 L_s^0 \frac{F_{ir}F_{is}}{n} \right)$$

If we suppose that the two emitting patches have the same luminosity, $L_r{}^0$, then

$$Var\left(L_i^1\right) = R_i^2 \left(L_r^0\right)^2 \frac{\left(F_{ir}+F_{is}\right)\left(1 - \left(F_{ir}+F_{is}\right)\right)}{n}$$

and from the point of view of patch i, there is only one *virtual* patch, with the form factor sum of the components and the error behaving accordingly. We will use this result in section 4.

The only bound for the error in radiosity we have so far encountered is from Shirley [Shirley91], and specifically, is a bound for the variance, or Monte Carlo error. And to establish the bound we need to know the maximum of radiosity in scene, that is, it is *a posteriori* bound, while in ours it is *a priori*   bound.

3.4 Further remarks about the error

We have to make an important distinction between local or form-factor errors and global or luminance error in Monte Carlo methods. We will show an example where the global error diminishes in the Monte Carlo method (more specifically, in our Monte Carlo method), but enlarges in any other method. We suppose that in other methods the error in form factor is a biased one, that is, there is always an error by default or by excess (see for instance the error curves in [Wallace89]). Taking into account this supposition, and to get the same final precision, we need to have the values of form factors in non Monte Carlo methods with an accuracy at least three times that of a Monte Carlo one.

Take a cubical box, all the form factors are 0.2. There are about 1000 lines through every face, which makes a probable error (that is, $\alpha=0.5$) in the form factor of 0.0085. Take now a form factor of 0.197, which makes an error of 0.003, that is, about one third of the probable error. Suppose all form factors are 0.197. On the other side we have a matrix of Monte Carlo form factors. Let the reflectances equal {0.5,0.3,0.5,0.2,0.3,0.3}, and the initial luminances {0,0.5,0,1,0,0}. We are going to compare the results of the first iteration for the two given form factors and the results for the real ones (0.2).

We have in table 6 the results of the first iteration with the corresponding relative error, and the converged solution also with relative error:

Patch	Method	First iteration	Relative error	Converged solution	Relative error
1	Proposed	0.142487	5.00%	0.206658	3.26%
	Other	0.14775	1.5%	0.208987	2.17%
	Exact	0.15	-	0.213629	-
2	Proposed	0.5591	0.16%	0.602331	0.39%
	Other	0.5591	0.16%	0.602156	0.42%
	Exact	0.56	-	0.604712	-
3	Proposed	0.144405	3.73%	0.208027	2.62%
	Other	0.14775	1.5%	0.208987	2.17%
	Exact	0.15	-	0.213629	-
4	Proposed	1.02096	0.094%	1.05199	0.007%
	Other	1.0197	0.023%	1.05044	0.14%
	Exact	1.02	-	1.05192	-
5	Proposed	0.0893072	0.77%	0.131043	1.48%
	Other	0.08865	1.5%	0.130057	2.22%
	Exact	0.09	-	0.133014	-
6	Proposed	0.088517	1.65%	0.130382	1.98%
	Other	0.08865	1.5%	0.130057	2.22%
	Exact	0.09	-	0.133014	-

Table 6. Results of the first iteration with the corresponding relative error, and the converged solution also with relative error.

Now the remarkable thing is that in non-Monte Carlo methods the error *always* increases, but in our Monte Carlo method it seems to smooth, which makes big errors decrease and small ones increase. Also note that with an initial (local) error of about three times, the solution in Monte Carlo is of the same quality. When we try exceeded form factors (for instance the value 0.203 for all form factors) we obtain the same result. Had we used ponderated form factors (defined in section 2), we would see that to obtain the same quality with a biased method, we would have needed about *five* times the accuracy.

4. COMPLEXITY

We show in this section that, to the same error bound, the cost is lineal with the number of patches.

First, we have the cost of creating a random ray, let's say kr. Second, the ray must be matched against every surface in the scene for intersection (suppose we make no use here of coherence). If this cost is ks, and ns is the number of surfaces, we have for every ray $ks\ ns$. We include here the cost of picking up the intersected patch, which is constant because we divide the surface in equally spaced patches.

Next, we must order the intersections list of the ray and increment the array of the intersections. The average number of intersections for a given ray is computed in this way: if nr is the total number of rays cast, then for any body i there will be *2 nr Si/Ssphere* intersections, because *Si/Ssphere* is the probability of intersecting the body, and as the bodies are convexes they are always intersected at two points. Then we sum for all bodies and divide by nr, to get *ni*

$$2 \sum_i \frac{S_i}{S_{Sphere}}$$

If we consider only the rays intersecting the walls , that is, if we do not include the visibility lists of zero length, we have *2 nr* intersections for the walls, and *2 nr Si/S$_{walls}$* for each body within the enclosure. So the average length of non null lists would be *2+ Σ Si/S$_{walls}$* where the sum doesn't include the walls.

Then the average cost of sorting the list of intersections is bounded by *ki ni log(2n$_b$)* the length of each list is bounded by *2 n$_b$*, because each line can at most intersect all the bodies in the scene (*n$_b$* bodies).

Finally, the cost of accessing the array is *ka ni* .

Then, summing up, we have

Cost of one ray *< kr + ks ns + ki ni log(2n$_b$) + ka ni*

and the total cost is, if *nr* is the number of rays

Cost *< nr (kr + ks ns + ki ni log(2n$_b$) + ka ni)*

Since, in principle, the cost is not dependent on the number of patches, we could have a big number of patches (if memory allows). Unfortunately, the greater the number of patches, the less the number of intersections per patch, and the bigger the local errors in form factors. We must therefore consider the increment needed in rays cast (nr) to keep the same amount of error.

We will see how the complexity of the proposed method is, for a given scene and a given error, lineal in the number of patches. That is, to get the same accuracy, if we double the number of patches, we must double the number of rays (we have to remark here that if we had to compute explicitly the form factor array, dividing every term of every visibility list by the corresponding term of the list of totals of lines, we would introduce a quadratical term. But we can delay this division until we solve the system).

Suppose we have patch i, and it is divided into equal patches i1 and i2. Suppose also that luminous patch j is divided into j1 and j2. If an average of n rays pass through i, then an average of *n/2* will pass through i1. Then, for the variance in L^1_{i1} we have two terms coming from the patches j1 and j2, and with the same luminance, L_j^0 . But as we saw in section 3, the error is the same as that of a single patch with the same luminance, that is, patch j . We should have, then, the same variance as before for patch i were it not for the fact that now only half the lines intersect patch i1. Then, if we double the lines cast we have the same variance.

$$Var\left(L^1_{i1}\right) = R_i^2 \left(L_j^0\right)^2 \left(\frac{F_{ij1}\left(1 - F_{ij1}\right)}{\frac{n}{2}} + \frac{F_{ij2}\left(1 - F_{ij2}\right)}{\frac{n}{2}} - 2 \frac{F_{ij1} F_{ij2}}{\frac{n}{2}} \right)$$

and then

$$Var\left(L^1_{i1}\right) = R_i^2 \left(L_j^0\right)^2 \frac{\left(F_{ij1} + F_{ij2}\right)\left(1 - \left(F_{ij1} + F_{ij2}\right)\right)}{\frac{n}{2}}$$

But $F_{ij1} + F_{ij2} = F_{ij}$, and if we double the lines cast

$$Var\left(L^1_{i1}\right) = R_i^2 \left(L_j^0\right)^2 \frac{F_{ij}\left(1 - F_{ij}\right)}{n}$$

that is , the same as the original

$$Var\left(L^1_i\right) = R_i^2 \left(L_j^0\right)^2 \frac{F_{ij}\left(1 - F_{ij}\right)}{n}$$

were it not for F_{i1j} in place of F_{ij}. Now, if we want compare F_{i1j} with F_{ij} we must take into account the relation $F_{i1j} + F_{i2j} = 2 F_{ij}$. (As the two patches are equal, if r_{i1j} are the lines from i1 to j, r_{i1} the lines through i1, and r_i the lines through i, we have $r_{i1j} + r_{i2j} = r_{ij}$, also $r_{i1} = r_{i2} = r_i/2$. Then $F_{i1j} + F_{i2j} = (r_{i1j} + r_{i2j})/r_{i1} = r_{ij}/(r_i/2) = 2 F_{ij}$).

If F_{i1j} were equal to F_{i2j}, the two variances above would be equal. If not, we can make the assumption that F_{i1j} is of the same order as F_{ij}, and then the variances are of the same order. And we could have a gain on variance of F_{i1j}, but a loss on variance of F_{i2j}. More specifically, if we use the quadratical error $E^2{}_1$ as a measure of the global error in the first order iteration (or as the first order global variance, an informal term) then

$$E_1^2 = \frac{\sum_i A_i \, Var\left(L_i^1\right)}{\sum_i A_i}$$

we see that the term

$$A_i \, Var\left(L_i^1\right) = A_i \, R_i^2 \left(L_j^0\right)^2 \frac{F_{ij}\left(1 - F_{ij}\right)}{n}$$

will be substituted by the two terms

$$A_{i_1} \, Var\left(L_{i_1}^1\right) + A_{i_2} \, Var\left(L_{i_2}^1\right) = \frac{A_i}{2} \, R_i^2 \left(L_j^0\right)^2 \left(\frac{F_{i1j}\left(1 - F_{i1j}\right)}{n} + \frac{F_{i2j}\left(1 - F_{i2j}\right)}{n}\right)$$

We have

$$F_{i1j}\left(1 - F_{i1j}\right) + F_{i2j}\left(1 - F_{i2j}\right) = F_{i1j} + F_{i2j} - \left(F_{i1j}^2 + F_{i2j}^2\right)$$
$$= 2 F_{ij} - \left(F_{i1j}^2 + F_{i2j}^2 + 2 F_{i1j} F_{i2j} - 2 F_{i1j} F_{i2j}\right) = 2 F_{ij} - \left(4 F_{ij}^2 - 2 F_{i1j} F_{i2j}\right)$$

and dividing by 2 (the denominator of A_i)

$$F_{ij} - \left(2 F_{ij}^2 - F_{i1j} F_{i2j}\right) = F_{ij} - F_{ij}^2 - \left(F_{ij}^2 - F_{i1j} F_{i2j}\right)$$
$$= F_{ij} - F_{ij}^2 - \left(\left(\frac{F_{i1j} + F_{i2j}}{2}\right)^2 - F_{i1j} F_{i2j}\right) = F_{ij} - F_{ij}^2 - \left(\frac{F_{i1j} - F_{i2j}}{2}\right)^2$$

In conclusion

$$A_{i_1} \, Var\left(L_{i_1}^1\right) + A_{i_2} \, Var\left(L_{i_2}^1\right) - A_i \, Var\left(L_i^1\right) = -A_i \, R_i^2 \left(L_j^0\right)^2 \left(\frac{F_{i1j} - F_{i2j}}{2}\right)^2$$

and that is the amount by which global error E_1^2 decreases. This decrease can be interpreted in this way : It measures how well the new division of the patches reflects the geometrical visibility of the scene, and the degree of error in supposing a constant radiosity over the old patch. We see, then, that doubling the number of patches, if we double the number of rays cast we get the same global error minus a term that measures how closely we are approaching the non constant radiosity function.

We devised a test: a cubical enclosure is divided first into six patches, one for each face, and after that into 54 equal patches, nine per face. For these geometries there are exact (analytical) form factors [Siegel81, Hsu67], and, therefore, exact solutions. We considered a face with initial radiosity 1, and reflectance 0.5, the other faces had no initial radiosity and a reflectance 0.6. As an estimator of the (final) variance, we took the quadratical error. The results are summarized in Table 7, and seem to confirm our theoretical findings.

scene	patches	average of lines per patch	variance
1	6	977	$4.941 \ 10^{-5}$
2	6	9870	$3.49519 \ 10^{-6}$
3	54	985	$3.65268 \ 10^{-5}$
4	54	10475	$2.9414 \ 10^{-6}$
5	54	104559	$3.50088 \ 10^{-7}$

Table7. Global variance in five test scenes.

From table 7 we can remark: 1) global variance seems inversely lineal to the lines per patch; 2) with more patches, to keep the same variance we need to keep the same number of lines per patch; 3) with more patches, with the same number of lines per patch, there seems to be a reduction of variance, as expected theoretically.

5. RESULTS

Figures 5 and 6 show two result images corresponding to the same scene with different reflectivities. The 24 surfaces of the scene were divided in a total of 1122 patches . There were two equal sources of light, one in the front wall (not seen), and the other in the left wall. We cast one million rays. The resulting lines per patch of wall were about 1400, per patch of the bigger cube about 900, and per patch of the other cubes about 600.

6. CONCLUSIONS

We developed a Monte Carlo technique to compute the form factors, based on Integral Geometry. In section 2 we presented three possible estimators and studied their efectiveness. We gave in section 3 a bound for luminances for whatever method we use in solving the radiosity equations system. We have shown how the local errors compensate each other in a certain way, and we gave the exact value of variance for the first order iteration. Also we showed how Monte Carlo methods do need not the same level of initial precision as other methods, and so are very competitive. In fact, we have found in our test figures that a level of acceptable quality is about 1000 lines per patch, far from expected if we take into account only local errors.

We proved in section 4 that the proposed method is at worst lineal for every possible geometry (of convex bodies and planar patches). Furthermore, we showed in what order the global variance or quadratical error decreases due to the finer discretization of the scene. Shirley [Shirley91] also gives a bound for the variance in Li, both first order (direct illumination) and final (with interreflections), but not a defined global variance. Shirley also states that for bounded local variance, that is bounded Var(Li) for all i, the expected number of rays traced is proportional to the number of patches (or zones in Shirley terminology). We give here the *exact* value for the first order variance. And we have found an additional lowering of variance, though we give it only for the first order global variance. Previous results need to know the radiosity solution to determine the error bound, because the bound is dependent on the maximum of radiosity in scene.

7. FUTURE WORK

The first future development must be to find a good adaptive refinement strategy suitable to the proposed algorithm; i.e. a technique which takes advantage of its features. This may be done from the relationship between the actual form factor and the "full" form factor obtained by means of "removing" occlusions.

The proposed technique deals with perfect diffuse environments. As is well known, this is often a too strong a restriction. So it will be extended in order to be applied to environments with diffuse and specular reflections without loss of efficiency.

This efficiency of the algorithm may be easily improved using several types of coherence as proposed for other techniques. Specifically, a good organization of the scene would increase the efficiency of the intersection function. This must be done for both the diffuse and the hybrid cases.

Finally, the method will be extended with minor changes to deal with parametrical non planar surfaces, both convexes and concaves.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[Baum89] BAUM, D.R., RUSHMEIER, H.E., WINGET, J.M., Improving radiosity solutions through the use of analytically determined form-factors . ACM SIGGRAPH Conf. Proc. (July 1988), pp. 325-334. Computer Graphics vol.23 num.3.

[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 . ACM SIGGRAPH Conf. Proc. (August 1988), pp. 75-84. Computer Graphics vol.22 num.4.

[Freund80] FREUND, J.E. , AND WALPOLE, R. E. ,*Mathematical Statistics*. Prentice-Hall International, Inc., London, 1980.

[Goral84] GORAL, C.M., TORRANCE, K.E., GREENBERG, D.P., AND BATTAILE, B., Modeling the interaction of light between diffuse surfaces. ACM SIGGRAPH Conf. Proc. (july 1984), pp. 213-222. Computer Graphics vol.18 num.3.

[Greenberg86] GREENBERG, D.P., COHEN, M.F., AND TORRANCE, K.E., Radiosity: a method for computing global illumination. The Visual Computer 2, 5 (1986), 291-297.

[Hanrahan91] HANRAHAN, P., SALZMAN, D., AUPPERLE, L. A rapid hierarchical radiosity algorithm. ACM SIGGRAPH Conf. Proc. (july 1991), pp. 197-206. Computer Graphics vol.25 num.4.

[Hsu67] HSU, CHIA-JUNG Shape factor equations for radiant heat transfer between two arbitrary sizes of rectangular planes. The Canadian journal of Chemical Engineering, vol. 45, February 1967, pp.58-60.

[Meshkov81] MESHKOV, V.V. *Fundamentals of Illumination Engineering* Editorial MIR, Moscú, 1981.

[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.

[Rey-Pastor51] REY PASTOR, J., SANTALO SORS, L.A. ,*Geometría Integral*, Espasa-Calpe, 1951

[Santaló76] SANTALO , L.A. ,*Integral Geometry and Geometric Probability*. Addison-Wesley, 1976

[Sbert93] SBERT, M., An Integral Geometry Based Method for Fast Form-Factor Computation Research Report. L.S.I. 93-4-R . Universitat Politècnica de Catalunya, 1993.

[Shamsundar73] SHAMSUNDAR,N., SPARROW, E.M., HEINISCH, R.P., Monte Carlo radiation solutions. Effect of energy partitioning and number of rays. Int.J. Heat Mass Transfer. Vol. 16, pp. 690-694.

[Shirley90] SHIRLEY,P. , A ray tracing algorithm for global illumination. Graphics Interface '90, May 1990.

[Shirley91] SHIRLEY,P. , Time complexity of Monte Carlo radiosity , pp. 459-465, Eurographics 91

[Siegel81] SIEGEL, R., HOWELL, J.R. ,*Thermal Radiative Heat Transfer* Hemisphere Publishing Corporation, Washington D.C., 1981

[Sobol76] SOBOL, I.M., *Método de Montecarlo Lecciones populares de matemáticas* .Editorial MIR, Moscú, 1976.

[Spencer90] SPENCER, S. N. The Hemisphere Radiosity Method: A Tale of Two Algorithms Eurographics Workshop on Photorealism in Computer Graphics. Rennes, 1990

[Wallace89] WALLACE, J.R., KELLS A. ELMQUIST, HAINES, E.A., A ray tracing algorithm for progressive radiosity ACM SIGGRAPH Conf. Proc. (july 1989), pp. 315-324. Computer Graphics vol.23 num.3.

[Weiner65] WEINER, M.M., TINDALL, J.W., CANDELL, L.M. Radiative interchange factors by Monte Carlo. Paper No. 65-WA/HT-51, ASME, November, 1965