Monte Carlo Approximation of Form Factors with Error Bounded a Priori*

M. Pellegrini[†]

April 16, 1996

Abstract

The exchange of radiant energy (e.g. visible light, infra-red radiation) in simple macroscopic physical models is sometimes approximated by the solution of a system of linear equations (energy transport equations). A variable in such a system represents the total energy emitted by a discrete surface element. The coefficients of these equations depend on the *form factors* between pairs of surface elements. A form factor is the fraction of energy leaving a surface element which directly reaches an other surface element. Form factors depend only on the geometry of the physical model. Determining good approximations of form factors is the most time consuming step in these methods, when the geometry of the model is complex due to occlusions.

In this paper we introduce a new characterization of form factors based on concepts from integral geometry. Using this characterization, we develop a new and asymptotically efficient Monte Carlo method for the simultaneous approximation of all form factors in an *occluded* polyhedral environment. The approximation error is bounded without recourse to special hypothesis. This algorithm is, for typical scenes, one order of magnitude faster than methods based on the hemi-sphere paradigm or on Monte Carlo ray shooting.

Let A be any set of convex non-intersecting polygons in R^3 with a total of n edges and vertices. Let ϵ be the error parameter and δ be the confidence parameter. We compute an approximation of each *non-zero* form factor such that with probability at least $1-\delta$ the absolute approximation error is less than ϵ. The expected running time of the algorithm is $O((\epsilon^{-2}\log\delta^{-1})(n\log^2 n + K\log n))$, where K is the expected number of *regular intersections* for a random projection of A. The number of regular intersections can range from 0 to quadratic in n, but for typical applications it is much smaller than quadratic. The expectation is with respect to the random choices of the algorithm and the result holds for any input.

*A preliminary version of this paper appeared in Proceedings of the *11th ACM Symposium on Computational Geometry*, pages 287–296, June 1995, Vancouver.

[†]Istituto di Matematica Computazionale del C.N.R. Via S. Maria 46, 56126 Pisa, Italy. tel: (+39) 50 593452 fax: (+39) 50 554342 e-mail: pellegrini@imc.pi.cnr.it

1 Introduction

1.1 Radiant Energy Transport in Heat Engineering

The determination of radiation interchange between surfaces is a central problem in heat transfer, illumination engineering an applied optics [SH92]. Complex radiation interchange problems are simplified using a variety of assumptions and approximations. One of the simplest models in heat transfer is that in which (i) each surface is "black" (i.e. there is no reflected radiation and all emitted energy is diffuse) and (ii) each surface emits diffuse radiation uniformly in each direction and from each point of the surface (see [SH92] for an extended treatment). The balance of energy in the model is described by a system of linear equations (energy transport equations). For a surface i in a set of n surfaces, let E_i be the emissivity (i.e. the energy produced by the surface i), ρ_i the diffuse reflectance (i.e. the proportion of received energy that is re-emitted) and B_i the total energy emitted by surface i. Denoting with F_{ij} the form factor between surface i and surface j, the energy transport equation for surface i is:

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j,$$

where the summation is extended to all surfaces different from i. All the transport equations form a linear system in the variables $B_1, .., B_n$. The values of ρ_i and E_i are known for a given model. The form factor (also called *configuration factor* or *view factor*) is defined as *the fraction of the diffuse energy leaving one surface that directly reaches another surface.* Under the conditions of the model, form factors depend only on the geometry of the surfaces and are independent from all the other parameters of the model. The determination of the form factor is an essential preliminary step to the solution of the system.

1.2 Radiosity Problems in Graphics

In graphics, the goal of synthesizing realistic images has been pursued by computing the balance of light reaching each surface. Such approach is known as *radiosity computation.* ACM Siggraph and Eurographics annual conferences have sessions devoted to radiosity computations, and since 1990 Eurographics has organized annual workshops on rendering, featuring many papers on radiosity. The Siggraph on-line bibliography lists more than 240 titles including the word "radiosity" in the period from 1983 to 1995. Some papers often cited in the literature are: [GTGB84] [NN85] [CG85] [Kaj86] [WRC88] [WEH89] [SP89] [HSA91] [Mit91] [Shi91]. For more recent results we refer the reader to the proceedings of specialized conferences and workshops (e.g. [HP95, BJ94, TT93]).

When we consider only the effect of diffuse light we can use assumptions similar to those used in the study of heat transfer and we obtain the same model described above (in graphics known as Lambertian model). Again, the determination of form factors is a central computational problem.

Exact analytical computation of form factors is difficult except for restricted special cases, none of which considers occlusions. References to analytic solutions for special cases or to computer programs for general cases can be found in the book of Siegel and Howell [SH92]. Several approximation schemes have been devised to cope with realistic environments.

1.3 Current approximation algorithms

There are currently several methods available for computing approximations of form factors. *Monte Carlo ray tracing* is chronologically the first method proposed. This method is still much used and researched because of its flexibility and ease of implementation. The method is based on simulating the exchange of energy by performing ray-shooting operations. One of the most popular schemes for a direct estimation of form factors is the *hemicube algorithm* [CG85], which is an elaboration of the hemi-sphere method of Nusselt [SH92]. Hanrahan et at. [HSA91] have proposed methods for computing form factors based on defining each form factor as a matrix of sub-form-factors and on an efficient hierarchical block decomposition of such matrices.

A detailed description of these and other algorithms is beyond the scope of this paper. We postpone some considerations to subsections 1.8 and 1.9. In Appendix A we give some more details while we attempt to compare these methods with the result presented in this paper.

1.4 New results in this paper

Within the discipline of design and analysis of algorithms, approximation methods are compared according to two main measures: one is the computational resources (time, storage) used, the second is the quality of the approximation, i.e. a bound on the error of the approximation. The two measure are not independent in general. Until now the form factors problem has been unyielding to this type of analysis. In this paper we present the following results:

(1) We give an integral geometric interpretation to the form factors using the style and notation of Santalo [San76] and we show that the analytic and geometric formulations previously used are *specializations* of this formula.

(2) We derive an efficient Monte Carlo algorithm to compute approximations of form factors for which we have *simultaneously* an asymptotic expected upper bound on the running time *and* an exact *a priori* upper bound, which holds with high probability, on the absolute approximation error of each form factor. The algorithm has better asymptotic running time, for typical scenes, than other methods for which a comparable analysis of the running time is possible (in particular, Nusselt's hemisphere method, and Monte Carlo ray tracing).

More precisely the algorithmic result in this paper is the following. Let A be a set of convex non-intersecting polygons in R^3 with a total of n edges and vertices[1]. We fix an error bound ϵ and a confidence δ. Let $K_r(A, v)$ be a certain subset of the intersection points

[1] We can think of these polygons as forming a covering of the facets of the input polyhedra.

among the orthogonal projections of the edges of A in direction v, which we call, following Mulmuley, *regular intersections*. We denote with K the expected size of $K_r(A, v)$ over a choice of v uniform at random in the set of all directions.

Theorem 1 *We can compute in expected time $O((\epsilon^{-2} \log \delta^{-1})(n \log^2 n + K \log n))$ an approximation of each non-zero form-factor such that with probability $1 - \delta$ the absolute error on each form factor is less than ϵ.*

The number of regular intersections can range from 0 to quadratic in n, but for typical applications it is much smaller than quadratic. A typical input is a set of polygons in which polygons are rather fat or, to put it negatively, not long and thin. This is a typical input because in these conditions the assumptions of the Lambertian model are more likely to be realistic.

1.5 What is a form factor?

The first task we set out to accomplish is to give a simple definition of a form factor in the style and notation of Santalo [San76]. The connection between the measure of sets of lines and form factors is the basis behind any derivation of the form factors integral. Here we apply this idea in a way fundamentally different form previous known derivations. The original definition of a form factor, as the fraction of diffuse energy leaving a surface that reaches directly another surface, belongs more to physics than to geometry. A further elaboration [SH92] equates the form factor to a double area integral in which the kernel function is diverging thus creating problems of numerical instability. Nusselt gives a geometric interpretation of a form factors in terms of central projections on the surface of a sphere, which holds only when one of the two surfaces is infinitesimally small (a so called differential area). Alternatively, researchers have used the connection between measure of lines and form factors to develop sampling strategies in spaces of rays or lines.

The author whose starting point is closest to the one used in this paper is Sbert [Sbe93] (see also [SPNP94, SPP95]). After noting that Lambert's cosine law is equal to one of the expressions for the differential element of lines, Sbert interprets the form factor as a geometric probability in a space of lines. Then a method is proposed based on choosing a set of lines randomly and counting the intersections between lines and the objects.

Starting from first principles, we show that form factors can be interpreted as the ratio between the measure of certain sets of lines in 3-space, *which is independent of any particular parametrization of lines*. We then show that the characterization of form factors previously used in literature are equivalent to our "abstract" characterization for specific choices of line coordinates. The benefit of starting from first principles is that proving the correctness of the new integral expressions of form factors becomes almost trivial within the framework provided by Santalo's exposition of integral geometric theory.

1.6 A new general method.

Our definition of form factor via integral geometric theory is abstract in the sense that it is not immediately computable and also in the sense that it is independent of any coordinates of lines in 3-space. When we decide for specific coordinates of lines we obtain an equivalent formulation that is, in principle, computable.

The next step in our research hinges on the choice of coordinates of the lines so that the resulting concrete formulation for the form factor contains quantities that can be computed easily and efficiently using computation geometry techniques [PS85]. In our case we use only orthogonal projections and the area of planar convex polygons, for this reason we refer to our method as the *orthogonal projection method.* This formulation involves an integral which is simpler than those previously known.

1.7 An efficient algorithm.

We use Monte Carlo integration to approximate the value of the integral defining the form factor as derived by the previous analysis. The integration is based on sampling on the *unit sphere of directions*. This domain has only two parameters and has a bounded measure. These are important properties that we use to derive an exact bound on the variance of the integrand function and consequently a bound on the absolute error of the Monte Carlo integration.

For the case of two polygon, the algorithmic techniques needed are standard and can be found in [PS85]. For the case of many polygons, we use the vertical cylindrical decomposition of Mulmuley [Mul91] to compute efficiently the terms of the summations in the Monte Carlo approximation of the integral. Such data structures are built via plane sweep and dynamic maintenance of polygonal planar maps.

1.8 On the difficulty of comparing algorithms

Comparing the result presented in this paper with standard practice in graphics is difficult. The main contribution of our work on the theoretical side is a provable correlation between error and running time within the standard model used in Computational Geometry. An additional benefit is that the algorithm is almost linear in the number n of input vertices and in the average number K of regular intersections among edges.

Error analysis published in the graphics literature at my knowledge use additional assumptions, sometimes justifiable on pragmatic grounds, but usually avoided in computational geometry. For example, a widely referenced paper in the computer graphics community by P. Shirley [Shi91] proves that a straightforward Monte Carlo radiosity computation requires a number of rays *linear* in the number of objects present in the scene to attain estimates of the energy emitted by each surface within a predefined variance. This result is obtained in a rather strong model for which, among other assumptions, the total surface of the objects remains constant as the number of objects increases [Shi91, page 462]. No such assumption is used in this paper.

In [Sbe93] the input objects are enclosed in a sphere S and a certain number of random lines meeting the sphere is produced. The number of lines needed to obtain a predefined error bound with high probability on each form factor depends on the ratio between the area of the surface of the sphere and the smallest area of an object. A further case that is analyzed in [Sbe93] is one in which the number of patches increases as a result of a *subdivision* of the objects initially present in the scene. Our result does not depend on any hypothesis on the minimum area of the surfaces or on the diameter of the input; and it holds also when the number of objects increases, say, by addition of new objects to the scene.

Most of the currently known algorithm are hard to analyze and therefore the correlation between running time and approximation error has been investigated by means of experiments, rather than algorithmic analysis. Another approach has been to produce error analysis *a posteriori*, that is an error analysis based on values computed by the algorithm itself [LSG94].

In Appendix A we attempt the following type of comparison. We translate some of the popular methods within the model used by our algorithm. Whenever possible we try to use exact computations instead of non-exact computations whose contribution to the error is not quantified.

1.9 On the standard Monte Carlo ray tracing technique

In the standard Monte Carlo ray tracing technique we send random rays from a surface and we count the number of such rays reaching a second surface. The ratio between the two numbers is an estimator of the from factor relative to the two surfaces.

It is easy to prove that for a predefined level of error and confidence the needed number of such rays is constant per surface (see e.g. [Sbe93]), and roughly proportional to ϵ^{-2}. Thus to estimate all the form factors we need to trace $O(\epsilon^{-2}n)$ rays among n surfaces. The complexity of ray tracing is still an open field of investigation. Currently the best asymptotic bounds in exact analytical models for ray tracing require roughly $O(m^{0.8}n^{0.8})$ operations to trace m rays among n triangles [Pel93, AM93]. These methods, besides being quite complex and mostly of theoretical interest, are not known to exhibit a dependence on other combinatorial parameters of the input, such as the number K of regular intersections. An interesting feature of our algorithm is that, still in an exact model, we are able to derive a simple algorithm whose running time is almost linear in n and K.

From a more practical point of view, it is well known that, as a rule of thumb, sampling in a lower dimensional space reduces the variance of an estimate by a Monte Carlo method [BGS$^+$66]. In the case of a straightforward Monte carlo ray tracing where the rays are sampled uniformly among those meeting the first object, the variance of the associated random variable is *equal to* $F_{12} - F_{12}^2$. The expression for the variance in our method, given in (11), is *bounded* by $2F_{12} - F_{12}^2$, and is much smaller than such bound in certain cases. For example, when the area of the second surface is less than half of the area of the first surface the variance of our method is smaller than the variance of straightforward Monte Carlo ray shooting. My conjecture is that for values of the form factor below a threshold the variance of our method is always better than the variance of the ray shooting method.

1.10 Organization of the paper

In Section 2 we give the two characterization of form factors used most frequently in literature. In Section 3 we introduce an "abstract" characterization via integral geometry. In Section 4 we derive a new "concrete" integral definition of form factor. In Section 5 we derive the error bound on the Monte Carlo evaluation of such an integral. In Section 6 we give the algorithm to compute all form factors simultaneously. In Section 7 we speculate on possible extensions of our method to non-Lambertian models, where the total energy emitted by a surface is not uniform. In Appendix A we attempt a rough comparison between several methods for form factor computations.

2 Currently used concrete characterizations

The algorithms known in literature to compute form factors are mainly based on two formulations of form factors. In the first formulation (which is sometimes taken as the definition tout court) the form factor is a weighted double area integral [SH92]. Given a set of disjoint surfaces[2] $\mathcal{S}$ in 3-space, the form factor between two surfaces S_i and S_j in $\mathcal{S}$ is:

$$F_{ij} = \frac{1}{A_i} \int_{p \in S_i} \int_{q \in S_j} f(p,q) dp dq, \tag{1}$$

where:

$$f(p,q) = \frac{\cos\theta_i(p,q)\cos\theta_j(p,q)}{\pi |pq|^2} V(p,q,\mathcal{S});$$

A_i is the area of the surface S_i; $\theta_i(p,q)$ is the angle between the normal vector to S_i at point p and the line through p and q; $\theta_j(p,q)$ is the angle between the normal vector to S_j at point q and the line through p and q; $|pq|$ is the distance between point p and point q; and $V(p,q,\mathcal{S})$ is a predicate that has value 1 if the open segment pq does not meet any surface, and is 0 otherwise.

The second characterization due to Nusselt [SH92] defines the form factor between a differential surface of area dA_i and a finite surface A_j geometrically as follows. Consider the set of points of A_j visible from a point $p \in dA_i$ and the central projection from p of such set onto the surface of a sphere centered at p. Orthogonally project such set of points from the sphere onto the plane tangent to dA_i at p. The area of the set of points obtained from the second projection is the value of the (differential) form factor between dA_i and A_j. To obtain the value of the form factor between S_i and S_j we need then to integrate over S_i and divide by A_i. The analytic equivalent of Nusselt's formulation is:

$$F_{ij} = \frac{1}{A_i} \int_{p \in S_i} \int_{\omega \in \Omega} g(p,\omega) dp d\omega, \tag{2}$$

[2]In this paper we consider mainly polygonal surfaces, but most observations hold for a larger class of surfaces.

where:

$$g(p,\omega) = \frac{\cos\theta_i(p,\omega)}{\pi} V(p,\omega,S_j,\mathcal{S});$$

Ω is the set of directions; $\theta_i(p,\omega)$ is the angle formed by the normal to S_i at p and the ray from p in direction ω; $V(p,\omega,S_j,\mathcal{S})$ is a predicate whose value is 1 if the ray from p in direction ω meets S_j before any other surface in $\mathcal{S}$. An analytic closed form of Nusselt's characterization of the differential form factor for polygonal objects in 3-space was found by Lambert [Lam60] (see also [Arv94]).

3 An abstract characterization

3.1 Unoccluded case

We give an abstract characterization of form factors starting with the case in which we have only two polygons S_1 and S_2 and no occlusion between the them. Let us go back to the original intuitive meaning of form factor as *the fraction of the diffuse energy leaving one surface that directly reaches another surface*. In radiation models with non-participating mediums energy is transferred along linear trajectories, i.e. along lines. Thus we can measure the total energy $\mathcal{I}_1$ leaving one surface S_1 by summing the energy carried by each single line. Since there are infinite lines meeting a given surface S_1 the measure is more precisely defined by an integral:

$$\mathcal{I}_1 = \int_{L\cap S_1\neq\emptyset} I(L)dL, \tag{3}$$

where $I(L)$ is the density of the energy on the line L. Under the assumptions of the Lambertian model $I(L)$ is a constant $\bar{I}$. Thus we are left with a purely geometric integral to compute

$$\int_{L\cap S_1\neq\emptyset} dL = m(\{L : L\cap S_1 \neq \emptyset\}). \tag{4}$$

We introduce the notation $m(\mathcal{L})$ to denote such integral evaluated over the domain $\mathcal{L}$. It is convenient to look at Formula (4) using the theory of integral geometry as exposed in the book by Santalo [San76]. One of the concerns of integral geometry is measuring sets of geometric object (i.e. associating a positive real number to a set) in such a way that the measure is invariant under rigid transformations of the space. In our case we want the measure of the set of lines meeting the surface S to be invariant under rigid transformation of the Euclidean 3-dimensional space. The differential form dL denotes a differential for which this property of invariance holds. For any given parameterization of the lines in 3-space there is a corresponding formulation of dL that is unique up to constant multiplicative factors [San76]. Thus Formula (4) is nothing but the (invariant) measure of the set of lines meeting S_1.

The amount of energy $\mathcal{I}_{12}$ leaving a surface S_1 and reaching a surface S_2, if there are no occlusions, is given by the following integral over the lines meeting S_1 and S_2:

$$\mathcal{I}_{12} = \int_{L\cap S_1\neq\emptyset\wedge L\cap S_2\neq\emptyset} I(L)dL \tag{5}$$

Since $I(L)$ is constant and equal to $\bar{I}$ we are interested in the measure of the set of lines meeting both surfaces given by the integral:

$$m(\{L : L\cap S_1 \neq \emptyset \wedge L\cap S_2 \neq \emptyset\}) \tag{6}$$

The form factor F_{12} is given by the ratio of the two intensities (3) and (5):

$$F_{12} = \frac{\mathcal{I}_{12}}{\mathcal{I}_1} = \frac{\int_{L\cap S_1\neq\emptyset\wedge L\cap S_2\neq\emptyset} dL}{\int_{L\cap S_1\neq\emptyset} dL}. \tag{7}$$

From now on we use (7) as our integral geometric definition of form factor in absence of occlusions. From a classical result of integral geometry adapted to polygons [San76, page 246] we have that

$$m(\{L \in \mathcal{L} | L\cap S_1 \neq \emptyset\}) = \pi A_1$$

where A_1 is the area of S_1. Since we can easily compute the area of a convex polygon, we concentrate our attention on the numerator of the fraction in 7.
Formula (7) is valid for any coordinatization of the lines. In order to obtain a more concrete formula (as opposed to the somewhat abstract formulation in (7)) we need to decide what coordinates we use for parameterizing the line L. As a consequence we have a unique expression for dL as a differential form in the chosen coordinates.

It is easy to prove that, when we use as coordinates of a line L the two intersection points of L with the surfaces S_1 and S_2, Formulation (7) becomes Formula (1) [San76, page 230]. It is now clear that Formula (1) is linked to a particular choice of line coordinates. The hemisphere characterization of Nusselt can be derived from (7) when we use as coordinates of a line the intercept of L on a fixed plane (the plane tangent to dA_1) and the direction of L [San76, page 211]. In Section 4 we use a different parameterization of lines in 3-space and the correspondent invariant differential form to derive a new concrete formulation of (7).

3.2 Occluded case

Now we consider the general case in which we want to define the form factor F_{12} between S_1 and S_2 in presence of other objects $S_3, ..., S_k$. First of all, to simplify the situation, we notice that we need only consider the objects clipped within the convex hull of S_1 and S_2 since (portions of) objects outside $CH(S_1 \cup S_2)$ cannot possibly occlude a line meeting S_1 and S_2.

Therefore we want to compute the measure of the set of lines *meeting* S_1 and S_2, and *missing* all the clipped polygons $S_3^c, ...S_k^c$. Let $\mathcal{L}_{12} = \{L|(L\cap S_1 \neq \emptyset)\wedge(L\cap S_2 \neq \emptyset)\wedge_{i=3}^{k}(L\cap S_i^c = \emptyset)\}$ be such set of lines. The new abstract formula for the form factor is:

$$F_{12} = \frac{m(\mathcal{L}_{12})}{m(\{L|L\cap S_1 \neq \emptyset\})} \tag{8}$$

4 A new concrete characterization of form factors

4.1 Unoccluded case

Let us consider two disjoint convex planar polygons in 3-space, namely S_1 and S_2. From the abstract definition in (7) we just need to evaluate the integral:

$$m(\{L|L \cap S_1 \neq \emptyset \wedge L \cap S_2 \neq \emptyset\})$$

From Santalo [San76] we have that one of the possible parameterizations of the line L is by giving the plane $P(L)$ through the origin and orthogonal to L, and the intersection point $L \cap P(L)$. Using this coordinatization the form of the invariant density for lines in 3-space is $dL = d\sigma \wedge du$, where $d\sigma$ is the invariant density of points on a plane through the origin and orthogonal to L, and du is the invariant density of unoriented planes through the origin (equivalent to the points on the unit 2-dimensional sphere U centered at the origin where opposite points are identified).

We fix a point $u \in U$ and we integrate over the density $d\sigma$ of the plane through the origin and orthogonal to the vector $\vec{Ou}$, which we call $P(u)$. A line L orthogonal to $P(u)$ meets both S_1 and S_2 if and only if the intersection point $L \cap P(u)$ is in the intersection of the projection of S_1 onto $P(u)$ and the projection of S_2 onto $P(u)$. But the integral of the density of points in a given planar set is just the area of that set. So, denoting with $A_{12}(u)$ the area of the intersections of projections of S_1 and S_2 onto $P(u)$ we have that:

$$\int_{\{L|L\cap S_1 \neq \emptyset \wedge L \cap S_2 \neq \emptyset\}} dL = \int_{u \in U} A_{12}(u) du \tag{9}$$

Here we consider the case of two objects only and therefore there is no issue of occlusions. We can compute approximately the integral in Formula (9) by using Monte Carlo integration [BGS+66, DR75]. For a fixed direction u we can compute $A_{12}(u)$ in time $O(n)$, where n is the number of vertices of the polygons S_1 and S_2, for example by using a sweeping line approach. We choose uniformly at random on the unit sphere U a set of N points $u_1, ..., u_N$, then the Monte Carlo approximation we obtain is:

$$m(\{L|L \cap S_1 \neq \emptyset \wedge L \cap S_2 \neq \emptyset\}) \approx \frac{2\pi}{N} \sum_{i=1}^{N} A_{12}(u_i)$$

We can find an approximation to the form factor of two unoccluded polygons in 3-space in time $O(Nn)$. An upper bound on the error developed in section 5 for the occluded case holds also in the unoccluded case. Here we show the high level pseudocode of the algorithm to compute the approximate form factor between two polygons in 3-space without occlusions.

```
proc ff(A,B:3D-polygon; eps,de:real):real
begin
  N = floor(1/(eps * eps * de));
  ff = 0;
```

```
  for i = 1 to N do
    u = random-direction;
    A1 = projection of A along u;
    B2 = projection of B along u;
    ff = ff+(area(intersection(A1,B2));
  end for;
  return (2 * ff)/(N * area(A));
end
```

4.2 Importance sampling

We can improve on the basic idea of algorithm `ff` by filtering out efficiently the directions for which the contribution to the summation is zero. We sketch here the method. We assume, without loss of generality, that the plane spanning S_1 does not meet S_2 and viceversa[3]. Consider the convex hull $C = CH(S1 \cup S_2)$. Assuming general position, a facet f of C, except S_1 and S_2, is incident to an edge e of S_1 and to a vertex v of S_2, or viceversa. Moreover the plane spanning f leaves S_1 and S_2 on the same side. We find, using a sort of binary search, the plane incident to e that instead leaves S_1 and S_2 on *opposite* sides (we call *positive* the side containing S_1). We find such planes for every facet f and we translate them to the origin. We take the intersection of all the translated positive half-spaces. We obtain a double cone with apex at the origin, which we call, for lack of a better name, the *anti-convex-hull* of S_1 and S_2. It is easy to see that, for a projection direction u that falls outside the anti-convex-hull, the projections of the surfaces cannot meet. Instead, if the direction u falls within the anti-convex-hull, then the projections have a non-empty intersection. The convex hull and the anti-convex-hull can be built in time $O(n \log n)$. After preprocessing we can decide whether a sampled direction is inside the anti-convex-hull in time $O(\log n)$, since the problem is equivalent to locating a point in a planar convex polygon of $O(n)$ sides.

4.3 Occluded case

We compute the measure of the set of lines in Formula (8) by integrating the motion invariant unit element dL discussed in the previous sub-section. We indicate with S_i^c the surfaces clipped within $CH(S_1 \cup S_2)$. We indicate with the superscript $''$ to a set, the set obtained by orthogonal projection in direction u. We denote with A'_{12} the area of the set $(S_1'' \cap S_2'')/\bigcup_{j=3}^{k}(S_j^c)''$. The numerator of the fraction in (8) is thus equal to:

$$m(\mathcal{L}_{12}) = \int_{u \in U} A'_{12}(u)du. \tag{10}$$

In Section 6 we see how to compute the terms $A'_{ij}(u)$ simultaneously for all pairs ij for a fixed direction u. In the next section we give an upper bound to the error in a Monte Carlo approximation of the form factor using integral (10).

[3]This assumption can be eliminated easily by splitting each surface in at most two parts.

5 Error estimate

5.1 Bounding the variance

In order to bound the error of our method we use an elementary analysis of Monte Carlo methods as described in [BGS+66] and [DR75]. This type of analysis relies on a bound on the variance of the integrand function over the domain of integration. This bound is hard to obtain for the integrand functions used in previous methods for form factor computation. Luckily, in our case a bound on the variance is obtained very easily. We recall that the measure of the set U is 2π. We introduce a function $\zeta(u)$:

$$\zeta(u) = \frac{2A'_{12}(u)}{A_1}.$$

We have that the expected value E of $\zeta(u)$ over all directions U is the value of the form factor F_{12}:

$$\frac{1}{2\pi}\int_{u\in U}\zeta(u)du = \frac{1}{2\pi}\int_{u\in U}\frac{2A'_{12}(u)}{A_1}du = F_{12}.$$

Now we estimate the variance D of the function ζ over the set of directions U:

$$D(\zeta(u)) = E(\zeta(u)^2) - [E(\zeta(u)]^2 = \tag{11}$$

$$[\frac{1}{2\pi}\int_{u\in U}\frac{4(A'_{12}(u))^2}{A_1^2}du] - F_{12}^2 = \tag{12}$$

$$\frac{2}{2\pi}\int_{u\in U}(\frac{2A'_{12}(u)}{A_1})(\frac{A'_{12}(u)}{A_1})du - F_{12}^2 \leq \tag{13}$$

$$2F_{12} - F_{12}^2 \leq 1. \tag{14}$$

where we exploit the fact that $A'_{12}(u) \leq A_{12}(u)$ is always smaller than or equal to A_1.

5.2 A bound obtained using Chebyshev's inequality

For any random variable Y and $t > 0$, Chebyshev's inequality [Fel68, page 232] is:

$$Prob(|Y| \geq t) \leq E(Y^2)/t^2,$$

where $E()$ indicate the expected value. Let $S_N = \sum_{i=1}^{N}\zeta(u_i)$, $\mu = E(\zeta)$ and let σ^2 be the variance of $\zeta(u_i)$. We apply Chebyshev's inequality, obtaining:

$$Prob(|(S_N - N\mu)/\sqrt{N}\sigma| \geq t) \leq 1/t^2.$$

Rearranging and dividing by N, we obtain:

$$Prob(|(S_N/N - \mu)/| \geq t\sigma/\sqrt{N}) \leq 1/t^2.$$

Equivalently, if we call the absolute error ϵ, we have $\epsilon = t\sigma/\sqrt{N}$ and $t = \epsilon\sqrt{N}/\sigma$. The probability of exceeding the error bound is $\delta = 1/t^2 = \sigma^2/\epsilon^2 N$. The probability of being within the error bound is conversely $1 - \delta$.

In the previous section we proved $\sigma^2 \leq 1$, thus we can derive the number of samples attaining the desired precision as a function of ϵ and δ. We have: $N = \lceil 1/\epsilon^2\delta \rceil$. We have proved the following lemma:

Lemma 1 *The Monte Carlo evaluation of formula 10 results in an approximation of the form factor within an absolute error* ϵ*, with probability at least* $1-\delta$*, when we choose* $N = \lceil 1/\epsilon^2\delta \rceil$.

5.3 Reducing the number of samples

From the previous section we have that the number of samples needed for an (ϵ, δ) approximation algorithms is $N = \lceil 1/\epsilon^2\delta \rceil$. It is possible to reduce the number of samples to $N = O((1/\epsilon)^2 \log(1/\delta))$ by using a trick reported in [KKK+90], originally used in [JVV86].

For completeness we describe this method. We use the algorithm described in Section 6 with a constant value for the confidence parameter, say $\delta = 1/4$. We run the algorithm Q times, and we take the *median* value as the outcome of the algorithm for F_{ij}. We obtain an (ϵ, δ)-approximation by choosing $Q = c\log(1/\delta)$ for an appropriate constant c. We give a sketch of the proof in the rest of this subsection.

We call a value X_i good if it is with a distance ϵ from the expected value $E(X)$, and bad otherwise. In our experiment a sample will be bad with probability 1/4 and good with probability 3/4. If the median is bad, then there are at least $\lceil Q/2 \rceil$ bad samples. Therefore the probability of a bad median is less than the probability of having at least $\lceil Q/2 \rceil$ bad samples. To bound this second probability we use Chernoff bound in a form reported in [Spe87]. Let $Y_i = 1$ if X_i is bad and $Y_i = 0$ if X_i is good. We have that:

$$Prob(\sum_{i=1,Q} (Y_i - E(Y)) > a) < e^{-2a^2/Q}.$$

Using $E(Y) = 1/4$ and $a = Q/4$ we obtain:

$$Prob(\sum_{i=1,Q} Y_i > Q/2) < e^{-Q/8}.$$

Thus, to reach a confidence level δ we just need to repeat the algorithm $Q = O(log(1/\delta))$ times.

6 Measuring vertical visibility

The arguments in the previous sections reduce the problem of computing form factors to the problem of computing areas of intersections of planar polygons on an Euclidean plane. We compute such areas using an algorithm for vertical space partition as algorithmic skeleton. A good asymptotic performance is obtained using the cylindrical partition of Mulmuley [Mul91].

Let $\mathcal{S}$ be a set of interior disjoint convex polygons in 3-space and let $\mathcal{E}(\mathcal{S})$ be the set of edges of polygons in $\mathcal{S}$. We choose u as the vertical direction. Let q be an intersection among the orthogonal vertical projections of $e_1 \in \mathcal{E}(\mathcal{S})$ and $e_2 \in \mathcal{E}(\mathcal{S})$. Such a point is called a *regular* intersection point if the vertical line through q does not intersect any polygon of $\mathcal{S}$ between e_1 and e_2. If we call k_r the number of regular intersections and k the number of intersections, we have that $0 \leq k_r \leq k \leq n^2$. In typical situations, though, k_r will be much smaller than n^2. The vertical decomposition $H(\mathcal{S})$ for the set $\mathcal{S}$ is a decomposition of $R^3/\mathcal{S}$ into maximal vertical prisms[4] such that: (i) each prism does not intersect $\mathcal{S}$, (ii) the two bases of the prism are subsets of two polygons in $\mathcal{S}$.

Clearly, for every pair of polygons S_i and S_j, the value of A'_{ij} is obtained by summing the area of the projection of cylinders with bases in S_i and S_j. The cylindrical partition $H(\mathcal{S})$ has size $O(n + k_r)$ and is built in time $O(n \log^2 n + k_r \log n)$ where k_r is the number of *regular intersection points* [Mul91].

6.1 Overall algorithm

We summarize here the whole algorithm to compute form factors. C_{ij} is a counter for the pair of surfaces with indices $i < j$, which is initialized to 0. We organize such counters using an array indexed from 1 to n, holding pointers to dynamic binary search trees. The counters are at the leaves of the trees. Clearly we need to store only counters C_{ij} which, during the construction, are found to have a value different from 0. If a counter is not in the data structure its value is by default 0. In this way we can exploit the sparsity of the visibility structure and we avoid using quadratic storage if we do not need to. We use Q of such data structures.

The external loop is executed $Q = O(\log \delta^{-1})$ times. In the internal loop we select randomly and uniformly for $N = \lceil 4/\epsilon^2 \rceil$ times a plane through the origin (i.e. a vertical direction). We compute the vertical cylindrical decomposition of Mulmuley for that choice of the vertical direction. For each prism whose bases are supported by S_i and S_j, we add the area of the projection of the prism to the counter C_{ij}.

At the exit of the external loop we save the median value of the Q values for C_{ij}. When we want to compute the approximation of F_{ij} we access C_{ij} and we divide $2C_{ij}$ by NA_i. To approximate F_{ji} we divide by NA_j. We can find easily all non-zero approximate form factors by visiting the data structure. The size of the data structure is proportional to the number of non-zero approximations to form-factors.

6.2 Observations

(1) By choosing as value of the confidence parameter $n^2\delta$ we obtain that, with probability $1 - \delta$, *all* the non-zero form factors have a simultaneous good approximation. The running time is increased only by a factor $O(\log n)$.

[4] A prism here is a solid obtained by intersecting an infinite prism with two half-spaces.

(2) Our algorithm is particularly suitable in scenes with large emitting surfaces (e.g. a window). Sometimes scenes have small sources of large radiant emission. In this case our algorithm might estimate to 0 the form factor between this small source and small distant visible objects. On the other hand the fact that the source is small causes no trouble in computing form factors between the source and large close visible surfaces.
(3) It is possible to customize the algorithm so to compute only the form factors involving a specific polygon S_i by restricting the computation to the vertical cylinder based on S_i. The running time in this case would depend on the average number of regular intersections within the projection of S_i. This is desirable if we want to place more stringent error conditions on a form factor involving a bright surface S_i.
(4) Unlike methods based on Formula (1) our method does not have any divergence problem due to surfaces too close to each other.
(5) Our algorithm does not spend time in computing *zero* form-factors, thus it is particularly suited for scenes where the form factor matrix is sparse. Moreover, if a form factor is zero, its approximation is by default also zero.
(6) Implementing the algorithm for the case of two surfaces without occlusions should be easy. For the general problem, the difficult part is the computation of the vertical cylindrical decomposition. The 3-dimensional sweeping algorithm of Mulmuley, which gives us the best bound, is challenging to implement since it relies on the dynamic maintenance of planar maps [PT90]. If we are willing to relax the time bound we can instead project all the surfaces onto a plane and use as algorithmic skeleton the line sweep algorithm of Bentley and Ottman [BO79].

7 Conclusions and Extensions

In this paper we have presented a new method, or paradigm, for computing the form factors among polygons in 3-space with occlusions (christened *the orthogonal projections method*). We derive a bound on the absolute approximation error under the assumption that the integrand function is computed using exact real arithmetic in the RAM model. We describe a simple algorithm for computing an approximation to the form factor between two disjoint non-occluded polygons in 3-space. We describe an asymptotically efficient algorithm to approximate every non-zero form factor between disjoint polygons in an occluded 3-dimensional scene.

The integrals considered in this paper can be seen as associated with a Galerkin method where the functional basis is piecewise constant. When one uses a Galerkin method over a more sophisticated functional basis (e.g. Legendre polynomials, Jacobi polynomials and Wavelets) more complex integrals are obtained [Zat93, GSCH93, TM93, Hec92]. Usually approximate quadrature methods are invoked in order to estimate such integrals. In the context of our method one has to redefine the kernels of such integrals as a function of lines meeting pairs of surfaces, as opposed to a function of points on such surfaces. Such an integral could than be split in an external integral on a domain of directions and an inner integral associated with an orthogonal projection. If our surfaces are polyhedral and the functional basis is polynomial, such inner integrals are easy to compute exactly (see [Pel96] for such a technique applied in a different context). One possible advantage with respect to other methods is that dealing with occlusions without spoiling the quality of the approximation is easy within our framework.

8 Acknowledgments

Thanks Nina Amenta and the Geometry Centre at the University of Minnesota, where this research was initiated. Thanks to the anonymous referees whose comments have helped in improving the presentation.

References

[AM93] P.K. Agarwal and J. Matoušek. Ray shooting and parametric search. *SIAM J. on Computing*, 22(4):794–806, 1993.

[Arv94] J. Arvo. The irradiance jacobian for partially occluded polyhedral sources. *Computer Graphics*, pages 343–350, 1994. Proceedings of SIGGRAPH 94.

[BGS+66] N.P. Buslenko, D.I. Golenko, I.M. Sobol', V. G. Sragovič, and Ju. A. Šreider. *The Monte Carlo Method.* Pergamon Press, Oxford, 1966. Yu. A. Shrider (Ed.). Also *The method of statistical trials (The Monte Carlo method)*, Fizmatgiz, Moscow, 1962.

[BJ94] P. Brunet and F.W. Jensen, editors. *Proceedings of Eurogrpahics Workshop on Photorealistic Rendering in Computer Science.* Springer Verlag, 1994.

[BO79] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, C-28:643–647, 1979.

[CG85] M.F. Cohen and D.P. Greenberg. The hemicube: a radiosity solution for complex environemnts. *Computer Graphics*, 19(3):31–40, 1985. Proceedings of SIGGRAPH '85.

[DR75] P.J. Davis and P. Rabinivitz. *Methods of numerical integration.* Academic Press, New York, 1975.

[EJLA91] A.F. Emery, O. Johansson, M. Lobo, and A. Abrous. A comparative study of methods for computing the diffuse radiation viewfactors for complex structures. *Journal of Heath Transfer*, 113:413–421, 1991.

[Fel68] W. Feller. *An introduction to probability theory and its applications.* John Wiley & Son, New York, 1968. Third edition.

[GSCH93] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 221–230, 1993.

[GTGB84] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH '84 Proceedings)*, volume 18, pages 212–22, July 1984.

[Hec92] Paul Heckbert. Radiosity in flatland. *Computer Graphics Forum (Eurographics '92)*, 11(3):181–192, September 1992.

[HP95] P. Hanrahan and W. Purgathofer, editors. *Proccedings of the Fifth Eurographics Workshop on Rendering*. Springer Verlag, 1995.

[HSA91] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, 1991. Proceedings of SIGGRAPH '91.

[JVV86] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[Kaj86] James T. Kajiya. The rendering equation. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.

[KKK+90] N. Karmarkar, R. Karp, R. Karp, L. Lovasz, and M. Luby. A monte-carlo algorithm for estimating the permanent. Technical Report TR-90-063, Interantional Computer Science Institute, Berkeley, 1990.

[Lam60] Lambert. Photometria sive de mensura et gradibus luminis, colorum et umbrae, 1760. German translation by E. Anding in Ostwald's Klassiker der Exacten Wissenschaften, Vol 31-33, Liepzig, 1892.

[LSG94] D. Lishinski, B. Smith, and D.P. Greenberg. Bounds and error estimates for radiosity. *Computer Graphics*, pages 67–74, 1994. Proceedings of SIGGRAPH 94.

[Mit91] D.P. Mitchell. Spectrally optimal sampling for distribution ray tracing. *Computer Graphics*, 25(4):157–162, 1991. Proceedings of SIGGRAPH '91.

[Mul91] K. Mulmuley. Hidden surface removal with respect to a moving point. In *Proc. 23rd Annu. ACM Sympos. Theory Comput.*, pages 512–522, 1991.

[NN85] Tomoyuki Nishita and Eihachiro Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 23–30, July 1985.

[Pel93] M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9:471–494, 1993.

[Pel96] M. Pellegrini. Electrostatic fields without singularities: Theory and algorithms. In *Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 184–191, 1996.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry: an Introduction*. Springer Verlag, 1985.

[PT90] F. P. Preparata and R. Tamassia. Dynamic planar point location with optimal query time. *Theoret. Comput. Sci.*, 74:95–114, 1990.

[SAG94] B. Smits, J. Arvo, and D. Greenberg. A clustering algorithm for radiosity in complex environments. In *Proceedings of SIGGRAPH '94*, pages 435–442, 1994. In Computer Graphics.

[San76] L. A. Santaló. *Integral Geometry and Geometric Probability*. Addison-Wesley, Reading, MA, 1976.

[Sbe93] M. Sbert. An integral geometry based method for fast form-factor computation. In R. J. Hubbold and R. Juan, editors, *Eurographics 93*, pages 409–420, 1993. Also in Computer Graphics Forum, vol 12, num 3, 1993.

[SH92] R. Siegel and J.R. Howell. *Thermal Radiation Heat Transfer*. Hemisphere Pu. Co., Washington, 1992. Third edition.

[SH93] P Schroder and P. Hanrahan. On the form factor between two polygons. *Computer Graphics*, pages 163–164, 1993. Proceedings of ACM SIGGRAPH 93.

[Shi91] P. Shirley. Time complexity of monte carlo radiosity. In Werner Purgathofer, editor, *Eurographics '91*, pages 459–465. North-Holland, September 1991.

[SP89] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics*, 23(3):335–344, 1989. Proceedings of SIGGRAPH '89.

[Spe87] J. Spencer. *Ten lectures on the probabilistic method*, volume 52 of *CBMS-NSF Regional Conference series in Applied Mathematics*. SIAM, 1987.

[SPNP94] M. Sbert, X. Pueyo, L Neumann, and W. Purgathofer. Global multi-path monte carlo algorithms for radiosity. Technical Report IMA-94-04, Departament d'informatica i matematica aplicada, Universitat de Girona, 1994.

[SPP95] M. Sbert, X. Pueyo, and W. Purgathofer. Adding progressivity to the global monte carlo radiosity method. Technical Report IMA-95-01, Departament d'informatica i matematica aplicada, Universitat de Girona, 1995.

[TM93] Roy Troutman and Nelson L. Max. Radiosity algorithms using higher order finite element methods. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 209–212, 1993.

[TT93] N.M. Thalman and D. Thalman, editors. *Proceedings of Computer Graphics International '93*. Springer Verlag, 1993.

[WEH89] J.R. Wallace, K. A. Elmquist, and E.A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics*, 23(3):3115–324, 1989. Proceedings of SIGGRAPH '89.

[WRC88] G.J. Ward, F.M. Rubinstein, and R.D. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics*, 22(4):85–92, 1988. Proceedings of SIGGRAPH '88.

[Zat93] Harold R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 213–220, 1993.

A Comparison of algorithms for approximating form-factors

The algorithms developed for computing form factors come under many different families, some are randomized (Monte Carlo), some deterministic. A large variety of approximations in various aspects of the algorithm are used. Usually, neither the time bounds nor the error bounds are derived by analysis. For this reason we will not attempt to describe these methods as originally described in literature. Still, it is instructive to compare our method of orthogonal projections with the other known methods. The comparison is based on the following criteria:
(i) We consider the total cost of computing all form factors, that is, a form factor for each pair of surfaces, in presence of occlusions, for an input set of disjoint convex polygons in 3-space with n vertices and edges.
(ii) We classify algorithms into families according to the formulation of the form factor as an integral that is used as guideline of the computation.
(iii) We consider in each such family the Monte Carlo method in which the domain of integration is sampled uniformly at random and the integrand function is computed *exactly* in the real-RAM model.
(iv) For the exact computations of the integrand function we will refer currently known method with known worst case deterministic or expected time bounds.
(v) When we are not be able to bound the absolute error, we consider the less demanding requirement that the convergence rate of the error, as derived from simple analysis of the Monte Carlo method in [DR75] is at least $O(\epsilon)$ on each form factor. Here we rule out, for

example, the usual assumption done in the hemi-cube algorithm that the centre of a surface is sufficient to characterize form factors for the whole surface.

Many methods used in practice are obtained by those listed below by using approximations in determining also the integrand function and/or by using predefined lattice points instead of random points. A survey of methods used to compute diffuse radiation form factors is in [EJLA91]. The hierarchical decomposition method of Hanrahan et al. [HSA91] falls somewhat outside our Monte Carlo framework and the comparison of the running time of this algorithm with the running time of the method of orthogonal projections is more problematic.

1. **Analytical**. The double area integral is solved analytically. This method is used in simple case where occlusion is not present. For example, the method of Schröder and Hanrahan [SH93] for two unoccluded convex polygons gives the exact value in time $O(n^2)$ where n is the number of edges of the polygons. Occlusions are not treated and the closed forms used contain trascendental functions (bi-logarithmic function).

2. **Monte Carlo ray tracing**. See Introduction (Section 1.9).

3. **Central Projection Methods (or Hemisphere method)**. This method is based on Nusselt projection method [SH92] and is based on Formula (2). An exact computation of the integrand function in Formula (2) entails the computation of a spherical visibility map from a point. If we adopt an approach without preprocessing, we obtain a cost $O(n\epsilon^{-2} \times Cost(1\ visibility\ map))$. Computing visibility map may take time between linear and quadratic in n depending on the input. Thus the hemisphere method ranges in complexity between $O(n^2\epsilon^{-2})$ and $O(n^3\epsilon^{-2})$. More complex implementations involving preprocessing and/or output sensitive algorithms can improve the running time, although not in the worst case.

   The hemi-cube algorithm [CG85] is based on an approximate (discrete) computation of the visibility map using a rectangular grid and on using tabulated values of the kernel function in (2).

4. **Double area integration**. This method corresponds to a direct Monte Carlo integration of Formula (1). We choose randomly points on two surfaces S_i and S_j and, if the segment joining them does not intersect any other surface, we compute the integrand function in 1. The on-line approach to this method requires time $O(n^2\epsilon^{-2} \times Cost(1\ ray)) + Preproc(n))$. The considerations in Section 1.9 on ray shooting among polyhedra apply here as well.

5. **Hierarchical radiosity methods (HR)**. Hanrahan et al. [HSA91] describe a method for computing a matrix that approximate the form factor between two surfaces S_i and S_j. Surfaces S_i and S_j are subdivided recursively into sub-patches until the form factor between two patches drops below a (small) threshold value. The form factor between patches is bounded using the value of the differential form factor between the center of one patch and a disk enclosing the other patch, for which a simple closed formula is available. The gist of the argument is that not every pair of subpatches need to be

considered independently, instead, the resulting matrix of sub-form factors is structured into large rectangular blocks. Thus, even if we have expanded the computation of a single form factor into a (relatively) large matrix, the subsequent solution in terms of radiosity is speeded up by the block structure of the matrix. The correlation between the threshold values and the number of patches is studied experimentally. Similarly, the relation between the threshold value and the final error in estimating the form factor between the surfaces is explored by means of experiments.

Hierarchical radiosity methods of a second type deal with the problem of *clustering* the given surface into groups so to avoid computing all $O(n^2)$ form factors independently [SAG94]. Here the error analysis is of the a-posteriori type and is based on bounding the maximum value of the kernel function in (1) between well separated clusters of surfaces.

6. **Integral geometric characterizations**. See Introduction (Section 1.8).