A Geometric Approach to Computing Higher Order Form Factors (extended abstract)

Marco Pellegrini*

Abstract

The numerical solution of rendering and radiosity equations in computer graphics and heat engineering boils down to the repeated computation of certain 4- and 6-dimensional integrals known as form factors (FF), higher order form factors (HOFF) or rendering factors (RF). We consider the problem of computing any integral in this class defined over a pair of triangles in 3-space, without occlusions. An adaptive Gaussian integration scheme is described, which, in time $O(\log_c^2(1/\epsilon))$, computes any such integral within an absolute error ϵ, for a constant $c > 1$. This result is achieved by blending techniques from integral geometry, computational geometry and numerical analysis. In particular, integral geometry is used in order to obtain integrals whose kernel is free from singularities. Empirical testing confirms the fast convergence of the method. In particular we obtain relative errors in a large range $10^{-3} - 10^{-15}$, which is unusual among the competing methods.

1 Introduction

Rendering and radiosity equations have been widely used in computer graphics for photorealistic computerized rendering of sythetized scenes. Historically similar equations have been used earlier (and are still used) in heat engineering and radiant transport theory [27]. This paper describes an algorithm for approximating up to high precision a class of integrals that arise form a Galerkin-type numerical solution of rendering and radiosity equations. The integrals in question are known as *form factors* (FF), *higher order form factors* (HOFF) and *rendering factors* (RF).

*Istituto di Matematica Computazionale del CNR, Via S.Maria 46, 56126-Pisa (Italy). e-mail: pellegrini@imc.pi.cnr.it

0

1.1 New Results

We consider the case when the domain of integration is given by two triangles (or any constant size convex polygons) in 3-space without occlusions between them. On each triangle we may have an associated radiosity basis function which is a polynomial of fixed and small degree in Cartesian coordinates. We give an algorithm for which we prove that an absolute error ϵ is achieved in time $O(\log_c^2(1/\epsilon))$ for some $c > 1$ which depends on the input but not on ϵ (Theorem 1).

1.2 Comparison of theoretical bounds

For two convex polygons in 3-space in any relative position without occlusion between them an analytic formula for the FF is reported in [22]. Such an analytic formula does not extend to other integrals in the class we consider (HOFF and RF), moreover it is quite complex and does not allow for precision/time trade offs. Because of its complexity such formula is used mostly for benchmarking other (approximation) methods.

Monte Carlo methods such as those in [26, 18] give probabilistic time bounds roughly $O(\epsilon^{-2})$. As for methods based on Gaussian quadrature (e.g. [23] and [8, 9]), at the best of my knowledge no formal proofs on error bounds have appeared in the literature. As a rule of thumb for methods based on Quasi Monte Carlo integration or non-adaptive Gaussian integration applied to $\mathcal{C}^1$ functions, typically bounds roughly $O(\epsilon^{-1})$ are known.

1.3 Comparison of empirical results

The performance of numerical approximation methods is usually compared according to the number of functional evaluation (or computing time) versus relative error. We report both, although it should be kept in mind that computing power varies greatly among platforms and moreover the single functional evaluation in our method is quite different from that of other integration methods. The feature that we consider striking and less influenced by implementation issues is the large range of precision we attain which is much larger than those of competing methods. Moreover our range begins where that of most other methods ends.

Experimental results with our method are shown in Section 7 and figures 1, 2, 3 and 4. As test cases with known analytic solution we used two parallel axis aligned unit squares at several values of distance and two orthogonal unit squares sharing an edge. In both cases we achieve *relative error* in the range $[10^{-2} - 10^{-15}]$ in time ranging from 0.08 to 2 seconds of computing time.

Similar experiments reported in [4] with Monte Carlo, Hemicube and Malley's method give relative maximum error in the range $[10^{-1} - 10^{-2}]$. In [6] a relative error or 10^{-2} is considered the threshold of acceptability for applications in radiative heat transfer. Keller [16], using Quasi-Monte Carlo integration techniques in a similar experiment achieves a relative error in the range $[10^{-1} - 10^{-3}]$.

Issues raised by numerical approximation of integrals arising from higher order radiosity have been addressed in [23] and [8, 9]. Detailed experimental results on several radiosity methods are in [33]. Emphasis in these papers is on empirical estimate of the error in the solution of the radiosity discretized linear system rather than in the single coefficients. Thus it is not possible a comparison with the results presented in this paper which are targeted to the single entries of the radiosity matrix. Other Galerkin based radiosity methods in [34] [32] and [10] rely on standard four dimensional product Gaussian quadrature rules and no performance data is reported for the computation of a single HOFF.

1.4 Summary of the Algorithm

The algorithm in this paper and its analysis are derived from recent work of the author on approximation of integrals arising in the Boundary Element Method for electrostatics of conductors [19]. Transfer of concepts among different branches of applied mathematics is not unusual, however, we find striking that in this case we could find not only intuitive similarities but indeed a common theory based on integral geometry that unifies definition, algorithms and complexity proofs across areas (radiosity and electrostatics) which have developed separately. The algorithm in this paper and its analysis are slightly simpler than those in [19] because the functions involved are better behaved.

We start by extending the integral geometric formula for FF developed in [18] to include HOFF. In a companion paper [20] it is shown that also Rendering factors (RF) can be reduced to the same *normal form* (NF). The integrals in NF are 4-dimensional integral. We split the 4 dimensional integration into an external integration over the set of direction and an internal integration defined over the intersection of the projections of the triangles onto a common plane orthogonal to a given direction. For FF such inner integral is the area of the intersections, for HOFF and RF we have a more complex functional, which is however computable analytically when the basis functions are polynomials.

The integration over the sphere of directions is achieved through an adaptive product Gaussian quadrature rule[1]. Adaptivity is achieved by subdividing the sphere of directions into regions such that in each the integrand function is analytic. This property and classical results from numerical analysis [5] are key ingredients in the error analysis which establishes the time bound $O(\log^2(1/\epsilon))$. In my previous work [18] the integration over the set of directions was done by a Monte Carlo method thus resulting in a probabilistic algorithm with much weaker theoretical bounds and empirical performance.

[1] See [5] for background on Gaussian quadrature.

1.5 Organization of the Paper

In Section 2 we give background notions. In Section 3 we introduce the transformation of higher order form factors using integral geometric tools into a new formulation. Section 4 gives the main scheme of the proof of the result, recalling definition and theorems from [19]. In Section 5 we describe how we decompose the sphere of directions into suitable domains and the local system of coordinates in each domain. In Section 6 we terminate the proof of the main result. Section 7 describes the experimental results obtained by our method in computing FFs, for which exact reference values are available.

2 Background

Rendering equation. The rendering integral equation was introduced by Kajiya [15] to model the equilibrium of radiant energy exchange in a scene under a variety of local shading models including diffuse, glossy and specular reflectance within a a non-partecipating medium. Extended description of such equations can be found in books such as [7, 4, 28]. Using notation in [28] we define the total radiance function $L(y, \omega)$ to be the radiant energy emitted from a point y of the surfaces (we shall call Σ the collection of surfaces of the scene), in direction ω. Such function is given by the sum of several contributions. Energy might be emitted (as in light sources) by local phenomena, and this contribution is accounted by a function $E(y, \omega)$. Furthermore light received at y from all the other surfaces is re-emitted because of (different types of) reflection. The function $\rho(y, \omega, \eta)$ gives the fraction of the energy received at y from direction η that is re-emitted in direction ω. Calling n_y the external normal unit vector at y and $x(y, \omega, \Sigma)$ the first point of Σ met by a ray shot from y in direction ω, the rendering equation can be written as follows:

$$L(y, \omega) = E(y, \omega) + \int_{\eta \cdot n_y > 0} \rho(y, \omega, \eta) L(x(y, \eta, \Sigma), -\eta) \eta \cdot n_y d\eta \qquad (1)$$

The surfaces Σ and functions E, ρ are given are part of the input. Thus the objective is to find the function L which is the solution of (1). Analytic closed solutions are not known even in the simplest models so the continuous equation must be transformed into a system of linear equations in a finite number of real variables by the application of finite boundary or finite elements methods. Note that the function L has a four dimensional domain.

Radiosity equation. The radiosity equation is obtained from the rendering equation with the assumption that the function $\rho(y, \omega, \eta)$, also known as *bidirectional reflectance* *distribution function (BRDF)* is independent of the incoming and outgoing angles. From a physical point of view this restriction corresponds to retaining in the model effects due to perfect diffusion neglecting specular and glossy effects. Next we introduce the radiosity function $B(y)$ defined as the radiant energy emitted per unit of surface at point $y \in \Sigma$ (in any direction) and we obtain from (1) the following equation:

$$B(y) = E(y) + \rho(y) \int_{x \in \Sigma} B(x) G(x,y) V(x,y,\Sigma) dx \tag{2}$$

with

$$G(x,y) = \frac{\cos\theta_x \cos\theta_y}{|x-y|^2}, \tag{3}$$

where θ_x (resp. θ_y) is the angle formed by the outer normal n_x (resp. n_y) and the line joining x and y, while the function $V(x,y,\Sigma)$ has value 1 if the open segment xy does not intersect Σ, zero otherwise.

Again functions E, ρ and surfaces Σ are known and the objective is to find the function B which is the solution of (2). Again close form solutions are not known even in the simplest models and numerical finite element and finite boundary methods are used. Note that the domain of the function B is two-dimensional.

Galerkin discretization. At an high level the Galerkin method consists of two steps. In the first step the unknown function which is the solution of the integral equation is replaced by the linear combination of known functions taken from a finite set of functions called the *basis*. The elements of the basis are known, while the real coefficients of the linear combination are the variables to be determined. We can multiply both sides of any integral equation by any function (called a *test function*) defined on the same domain of the unknown function and then integrate over such a domain to obtain a new equation. The Galerkin method consists in taking as test functions exactly the basis. In the derived equations the only variables are the real coefficients of the linear combinations, therefore we can use linearity of integral operators to take them out of the integral sign. What is left is a linear equation where the coefficients are integrals involving the integral operator and the basis functions. For more details on finite element methods applied in radiosity radiosity we refer the reader to [13], [14] and [1].

Galerkin discretization applied to equation (2) leads to coefficients called *form factors* (FF) when the basis is made up of piecewise constant functions, or *higher order form factors* (HOFF) when the basis is not piecewise constant. Galerkin discretization applied to equation (1) leads to integrals which do not seem to have a standard name. We will refer to them in this paper as the *rendering factors* (RF).

Galerkin-type solution of radiosity equations with non piecewise constant basis have been introduced in computer graphics by Heckbert [12] for two-dimensional setting, by Zatz [34] and Troutman, Nelson and Max for the three dimensional setting [32]. The use of wavelet basis has been explored by Goertler et al. in [10]. The use of piecewise constant basis in numerical solution of radiosity equations has a longer history whose description is beyond the scope of this paper.

Handling occlusions. The visibility function $V(x,y,\Sigma)$ in equation (2) and the ray shooting function $x(y,\omega,\Sigma)$ in equation (1) are needed to model the phenomenon of occlusions. We have an occlusion when light traveling from a first surface to a second surface is intercepted by one or more opaque surfaces between them. What makes occlusions difficult to handle in computing FFs, HOFFs or RFs is the fact that it is not a local phenomenon since it depends on the whole set of surfaces Σ. There are four main approaches to handing occlusions.
The first approach uses explicit construction of visibility maps (either combinatorially or via Z-buffering) within the integration of either a single form factor or all of them in an interleaved way [3, 18, 17, 30].
The second approach is to approximate the integral by computing separately two terms: one is the FF (resp. HOFF and RF) where occlusions are disregarded, the second term is a correcting multiplicative factor involving only the occlusion function [11, 34, 29].
The third approach is to compute FF, HOFF and RF only when it is known, by use of auxiliary data structures (e.g. one in [31]), that the visibility function is one everywhere in the domain of interest [24, 25]. When this property is false or hard to establish the corresponding entry of the radiosity matrix is set to the default value zero.
The fourth approach is developed in a companion paper [20] and applies to (a modified version of) the rendering equation (1). Roughly speaking it consists in simulating occlusions by a superimposition of local light effects, so that no visibility predicate is passed to the definition of the rendering factors.

3 Geometric Transformations

In this section we start from the standard integral definition of Form factor and higher order form factor and we manipulate such integrals to obtain a new formulation. Given two disjoint surfaces (triangles) S_1 and S_2 in 3-space, the (normalized) form factor F_{12} between two surfaces S_1 and S_2 is [28, page 31]:

$$F_{12} = \frac{1}{A_1} \int_{p \in S_1} \int_{q \in S_2} G(p,q) dp dq, \tag{4}$$

where:

$$G(p,q) = \frac{\cos\theta_1(p,q)\cos\theta_2(p,q)}{\pi |pq|^2};$$

A_1 is the area of the surface S_1; $\theta_1(p,q)$ is the angle between the normal vector to S_1 at point p and the line through p and q; $\theta_2(p,q)$ is the angle between the normal vector to S_2 at point q and the line through p and q; $|pq|$ is the distance between point p and point q. In [18] it has been proved that the form factor between surfaces T_1 and T_2 in 3-space can also be represented as the following geometric integral:

$$F_{12} = \frac{1}{\pi A_1} \int_{u \in U/2} A_{12}(u) du, \tag{5}$$

where $U/2$ is half of the sphere of directions, A_1 is the area of the first polygon, and $A_{12}(u)$ is the area of the intersection of the projections of T_1 and T_2 onto a plane $P(u)$ of normal direction u.

Form factors arise when a piecewise constant approximation of the radiosity function defined over the surfaces in the scene is used to solve numerically rendering equations. If we use instead a non-constant functional basis, within a Galerkin-type approach, we obtain the *higher order form factors* (HOFF). Let us denote with $C_{1,h,2,k}$. the (unnormalized) HOFF between surfaces T_1 and T_2 associated with a local basis functions σ_h and σ_k. Using standard derivations [4, formula 3.39, page 60]:

$$C_{1,h,2,k} = \int_{p \in T_1, p' \in T_2} G(p,p')\sigma_h(p)\sigma_k(p')dpdp' \qquad (6)$$

where the functions σ_h and σ_k are known elements of the functional basis, possibly incorporating the diffuse reflectance function. Next we apply integral geometric transformations to the integral in (6). To simplify the notation we drop subscripts inherited from the discretization process and we consider two triangles T_1 and T_2 and any function σ_1 and σ_2 polynomial in Cartesian coordinates. Thus the integral we are considering is:

$$C_{12} = \int_{p \in T_1, p' \in T_2} G(p,p')\sigma_1(p)\sigma_2(p')dpdp' \qquad (7)$$

Now we can turn to a geometric integral since G is an expression of the differential measure of lines in 3-space [21] dL:

$$\int_{L, L \cap T_1 \neq \emptyset, L \cap T_2 \neq \emptyset} \sigma_1(p)\sigma_2(p')dL \qquad (8)$$

Next we express a line L in the (u,q) coordinates, where u is the direction of L and q the intersection of L with a plane $P(u)$ of normal u passing through the origin. In these coordinates the differential measure of lines is $dL = dudq$.

$$C_{12} = \int_{u \in U/2} \left[\int_q \sigma_1(p)\sigma_2(p')dq \right] du \qquad (9)$$

The domain of q in the inner integral is the intersection of the projections of T_1 and T_2 on $P(u)$. We have obtained an integral similar to integral (13) in [19], which arises in the context of the Boundary Element Method for Laplace's equation. The main difference is a term $1/\cos\phi$ which was present there, as a result proofs needed in this paper are mostly simplifications of those in [19].

A generalized form which is useful for the case of Rendering Factors is when the external integration is not on the whole set $U/2$ but on a spherical convex polygon $V \subset U/2$ (intersection of a constant size set of hemispheres).

$$C'_{12} = \int_{u \in V} \left[\int_q \sigma_1(p)\sigma_2(p')dq \right] du \qquad (10)$$

The main theoretical result in this paper is the following theorem:

Theorem 1 *Integrals (5), (9) and (10) can be approximated with an absolute error ϵ in time $O(\log_c^2(1/\epsilon))$, for some $c > 1$.*

4 Proof of Theorem 1

We give an high level sketch of the aspects of the proof related to the error analysis, while in Section 5 we give more details on the algorithmic aspects. From [19] we take the following definition:

Definition 1 *Integral:*

$$\int_{u \in U} f(u)du$$

is **well-behaved** *if it can be represented as:*

$$\sum_j \int_{u \in D_j} f_j(u)du,$$

where for each j, $D_j \subset U$; and the collection of the domains D_j is a partition of U. Moreover, for each j we can find a local system of reference LSR_j of parameters α_j and β_j such that:

1. *In LSR_j,*

$$D_j = I_{j,\alpha} \times I_{j,\beta} = [\alpha_{j,0}, \alpha_{j,1}] \times [\beta_{j,0}, \beta_{j,1}]$$

That is, D_j is mapped to a bounded rectangular domain in LSR_j.

2. *By substitution $f_j(u)$ is mapped to $f_j(\alpha_j, \beta_j)$ and we have the invariant differential element of directions $du = g_j(\alpha_j, \beta_j)d\alpha_j d\beta_j$.*
3. *For every $\bar{\alpha} \in I_{j,\alpha}$, the restriction $f_j(\bar{\alpha}, \beta_j)g(\bar{\alpha}, \beta_j) : I_{j,\beta} \rightarrow R$ admits an analytic extension in an open rectangle $Rect(\bar{\alpha})$ of the complex plane which contains strictly I_β.*
4. *Fore every $\bar{\beta} \in I_{j,\beta}$, the restriction $f_j(\alpha, \bar{\beta}_j)g_j(\alpha, \bar{\beta}) : I_{j,\alpha} \rightarrow R$ admits an analytic extension in an open rectangle $Rect(\bar{\beta})$ of the complex plane which contains strictly I_α.*

By set A strictly containing B we mean that A contains B and the boundary of A is disjoint from the boundary of B. We use the following Theorem whose proof is in [19]:

Theorem 2 (Theorem 2 in [19]) *A well-behaved integral can be approximated using $O(n)$ Gaussian points while achieving error $O(c^{-\sqrt{n}})$, for some $c > 1$.*

In Section 6 we will establish the following lemma:

Lemma 1 *Integrals (5), (9) and (10) are well-behaved.*

which, together with Theorem 2, completes the proof of Theorem 1.

5 Spherical Decomposition

In this section we show how to sub-divide the sphere and a specific choice of coordinates in each sub-domain. We thus establish the property 1 for the sub-domains of a well-behaved integral. We give a specific form to the differential measure on the unit sphere (property 2). Moreover we prepare the setting for the proof of property 3 and 4 to be carried out in Section 6.

5.1 Vertices of the Domains

Let us call (x, y, z) the coordinates of a point p in an absolute Cartesian reference system. And let $q = (X, Y)$ represent the coordinates of the point on a plane $P(u)$ of normal direction u. Consider the formula for the HOFF integral:

$$C_{12} = \int_{u \in U/2} \left[\int_q \sigma_1(q)\sigma_2(q)dq \right] du$$

Let T_1 and T_2 be triangles in 3-space, U the sphere of directions, $U/2$ half of the sphere of directions (technically we identify opposite directions). With $T_i(u)$ we denote the projection of T_i onto the plane $P(u)$. Let $\sigma_1(x, y, z(x, y))$ be the basis function associated with triangle T_1, and $\sigma_2(x, y, z(x, y))$ be basis function associated with triangle T_2. We consider the image of such functions on the plane XY, given by:

$$f_i(X, Y) = \sigma_i(x(X, Y), y(X, Y), z(x(X, Y), y(X, Y)))$$

for $i \in \{1, 2\}$. Thus the function we want to integrate in the inner integral is

$$f_{12}(X, Y) = f_1(X, Y)f_2(X, Y).$$

The domain of integration is the set $T_1(u) \cap T_2(u)$ which is a convex polygon on $P(u)$. Let us call

$$Vert(u) = \{(X_0, Y_0), ..., (X_k, Y_k)\}$$

the collection of its vertices in counterclockwise order.

Lemma 2 *The integral*

$$A_{12}(u) = \int_q \sigma_1(q)\sigma_2(q)dq \tag{11}$$

is expressible as a polynomial in the coordinates of points in $Vert(u)$ in the XY local reference system.

Proof. The objective is to express the inner integral as a function of the coordinates of points in $Vert(u)$. We use Green's formula on the plane stating that, for functions h and k

$$\int_D (\frac{\partial h}{\partial x} - \frac{\partial k}{\partial y})dxdy = \int_{\partial D} kdx + hdy \tag{12}$$

We set

$$k = -(1/2) \int f_{12}(X, Y)dX,$$

$$h = (1/2) \int f_{12}(X, Y)dY.$$

Thus $f_{12}(X, Y) = (\partial k/\partial X) - (\partial h/\partial Y)$. Let us consider the right hand side of Green's formula. The right hand side can be computed over a polygonal chain by summing the contribution of each segment, let us consider w.l.o.g the first edge. Switching to local parametric coordinate t

$$X = (X_1 - X_0)t + X_0$$

$$Y = (Y_1 - Y_0)t + Y_0$$

we can turn the right hand side of (12) into an integral in t and integrating in the range $[0, 1]$. What is left after the integration in t is a polynomial in the quantities X_0, Y_0, X_1, Y_1. ∎

5.2 Combinatorially Invariant Regions on the Sphere

The set $Vert(u)$ depends on u. In this section we show how to divide the sphere U into regions such that in each region $Vert(u)$ is combinatorially invariant.

Let us consider two triangles in 3 space, T_1 and T_2. If we change u continuously, $Vert(u)$ changes from one combinatorial fingerprint to another when the projection of three vertices is collinear. Let us call a *transition curve* the locus of points in U for which three specified vertices are collinear. Every such locus is a great circle on U. The collection of such great circles form an arrangement on U and every 2-dimensional cell of such an arrangement corresponds to a set of directions for which $Vert(u)$ is combinatorially invariant. Thus one such region R_j is bounded by arcs of great circles and is a "convex" polygon in spherical geometry.

Remark 1. Note that the number of regions depends only on the six vertices of the two triangles and it is thus a constant with respect to the error parameter. For two quadrilaterals we still have a constant number of regions to consider.

5.3 The Forbidden Curve and the Forbidden Vector

Points in $Vert(u)$ are of two types. They are either projections of vertices of the input triangles, or they are intersections of the lines spanning projections of edges of input triangles. Points of the first type are defined for any value of u. Points of the second type are not defined for those directions u such that segments are parallel in the projection. We have the following lemmas which give a characterization of such directions.

Lemma 3 *Let $p' \in Vert(u)$ be the intersection of the lines defined by pairs of points $(p'_1p'_2)$ and $(p'_4p'_5)$. Then p' is the projection of a point $p \in R^3$ whose coordinates are of the form:*

$$p_x = \frac{h_x}{w \cdot u} + k_x$$

$$p_y = \frac{h_y}{w \cdot u} + k_y$$

$$p_z = \frac{h_z}{w \cdot u} + k_z$$

where $w, h_x, h_y, h_z, k_x, k_y$ and k_z are polynomials in the coordinates of p_1, p_2, p_3 and p_4.

Proof. Omitted in extended abstract.

We will call w the *forbidden vector* for p' and the great circle $w \cdot u = 0$ the *forbidden curve* of p'. Next we state a few useful properties of regions

Lemma 4 *The closure of a region R_j does not intersect forbidden curves from vertices of its fingerprint set $Vert_j$.*

Proof. It follows from the fact that within the validity region the formula represents the integral of a bounded function over a bounded domain such a value does not diverge, therefore there can be no point in common between a regions and forbidden curves. ∎

The forbidden vector and the forbidden curve are needed in the error analysis of the algorithm, however they are not actually computed by the algorithm.

5.4 The Sub-domains

By a spherical polygon we mean a compact subset of the sphere whose boundary is the union of arcs of great circles.

From the discussion in the first part of this section we have that the triangles T_1 and T_2 in R^3 induce on the sphere of directions an arrangement of great circles $\mathcal{A}$. Each cell R_j of such an arrangement is a spherical polygon with the property that the representation of the area function A_{12} is combinatorially invariant. Next we show how to split each such cell R into either quadrangular or triangular spherical polygons. A few definitions:

Let e be an arc of great circle, we call $GC(e)$ the great circle spanning e. Let u_1 and u_2 be distinct points on the unit sphere, we call $SGC(u_1, u_2)$ the unique great circle incident to them. Let u_1 be a point on the unit sphere, by $OGC(u_1)$ we denote the locus of directions orthogonal to u_1, which is a great circle.

Next we show how to subdivide each region R_j into sub-domains (see Figure 5). Let us call u_C the intersection of the great circle $GC(e)$ and $OGC(v)$. Take u_C as the "cutting pole" for R_j by drawing all great circles SGC through u_C and the vertices of R_j. This set of great circles slices R_j into a possibly empty set of spherical quadrilaterals and one spherical triangle incident on v. Note that in our construction the angle between v and u_C is $\pi/2$.
To summarize we have two types of sub-domains: quadrangular and triangular.

5.5 Local Coordinate System on the Unit Sphere

Quadrangular sub-domains. Here we assume that the sub-domain $D = R_j$ is quadrangular, with edges e_1, e_2, e_3, e_4, where e_1 and e_3 are the result of the slicing and e_2, e_4 are portions of edges of D. Consider the intersection point u_D among the great circles $CG(e_2)$ and $GC(e_4)$. In this case the joining great circle of u_C and u_D, $GC(u_C, u_D)$, does not intersect the boundary of D since D is the intersection of a spherical wedge based on u_C and a spherical wedge based on u_D. A spherical wedge can be defined by an planar angle at the pole of the wedge. Let us consider the points u_C, u_D and the point u_{CD} that is orthogonal to both. the form a skewed triple of versors. Call u_{CDD} the point orthogonal to u_{CD} and u_D. Call u_{CDC} the point orthogonal to u_{CD} and u_C. We have two orthogonal reference frames $B = (u_{CD}, u_{CDC}, u_C)$ and $S = (u_{CD}, u_{CDD}, u_D)$. We introduce local versors and coordinate notation: Let B be a reference frame (i_B, j_B, k_B) were i_B points up, k_B to the left, and j_B towards the viewer.
Projecting from j_B we see on the projection plane the x_B and z_B axis, let us call α_B the angle that the projection of u forms with the (vertical) x_B-axis, and a_B is tangent so:

$$a_B = \tan \alpha_B = \frac{u_{Bz}}{u_{Bx}}.$$

Projecting form k_B we see the x_B and y_B axis, let us call β_B the angle that the projection of u makes with the x_B axis, and $TANB_B$ its tangent so:

$$b_B = \tan \beta_B = \frac{u_{yB}}{u_{xB}}.$$

Calling $H = \sqrt{1 + a_B^2 + b_B^2}$ we have:

$$u_{xB} := \frac{1}{H},$$

$$u_{yB} := \frac{b_B}{H},$$

$$u_{zB} := \frac{a_B}{H}.$$

Now we consider the slanted reference frame S. Let S be a reference frame (i_S, j_S, k_S). The x_S-axis points at the middle of the hemisphere. We obtain S by rotating B of an angle η using the x_B axis as axis of rotation. Therefore $i_B = i_S$. In matricial form:

$$\begin{pmatrix} i_S \\ j_S \\ k_S \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\eta & -\sin\eta \\ 0 & \sin\eta & \cos\eta \end{pmatrix} \begin{pmatrix} i_B \\ j_B \\ k_B \end{pmatrix} \quad (13)$$

Inverting the matrix:

$$\begin{pmatrix} u_{xS} \\ u_{yS} \\ u_{zS} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\eta & \sin\eta \\ 0 & -\sin\eta & \cos\eta \end{pmatrix} \begin{pmatrix} u_{xB} \\ u_{yB} \\ u_{zB} \end{pmatrix} \quad (14)$$

We define the angles with respect to the S-reference frame: Projecting from j_S we see in the projection plane the x_S and z_S axis, let us call α_S the angle that the projection of u forms with the (vertical) x_S-axis, and a_S is tangent so:

$$a_S = \tan \alpha_S = \frac{u_{Sz}}{u_{Sx}}.$$

Projecting form k_S we see the x_S and y_S axis, let us call β_S the angle that the projection of u makes with the x_S axis, and b_S its tangent so:

$$b_S = \tan \beta_S = \frac{u_{yS}}{u_{xS}}.$$

Dividing both sides of this identity

$$u_{yS} = (\cos\eta) u_{yB} + (\sin\eta) u_{zB}$$

by u_{xS}, we obtain:

$$b_S = (\cos\eta) b_B + (\sin\eta) a_B,$$

and therefore:

$$a_B = \frac{1}{\sin\eta}(b_S - (\cos\eta) b_B).$$

We choose as our local coordinates system for the sub-domain D the pair of parameters $(\alpha_j, \beta_j = (b_B, b_S)$. We use the above equality to re-express H and the components of u_B in B as a function of b_B and b_S.
Triangular sub-domains. Here we assume that D is a triangular sub-domain. Therefore it is adjacent to v and, using the notation of the previous case $u_D = v$. D is the

intersection of a spherical wedge based on u_C and a spherical wedge based on v. The great circle $GC(u_C, v)$ meets the boundary of D in v. On the other hand by construction v and u_1 are at an angle of $\pi/2$. Therefore reference frames B and S are obtained one from the other by a rotation of $\pi/2$. Consider angles α_B and β_B. Let the range for β_B be $[\beta_0, \pi/2]$ so that the range for b_B is $[\tan\beta_0, +\infty]$. We can always choose $0 < \beta_0 < \pi/2$. We take an angle $\gamma = \beta - \beta_0$ so that calling $e = \tan\beta_0$ and $c_B = \tan\gamma$ we have:

$$c_B = \frac{b_B + e}{1 - b_B e}.$$

The range of integration now is $c_B \in [0 \ldots \tan(\pi/2 - \beta_0)]$ and c_B does nor diverge in this interval (although b_B would). Inverting the relation:

$$b_B := (c_B - e)/(1 + e c_B) \tag{15}$$

Now we choose as our local coordinates on the sphere $(\alpha_j, \beta_j) = (a_B, c_B)$. We can now express H, and the components of u_B as a function of a_B and c_B.

5.6 The Differential Measure of Directions in Local Coordinates

From Santalo [21] we take this form of the differential element of directions:

$$du = \frac{du_{zB} \wedge du_{yB}}{u_{xB}} \tag{16}$$

Quadrangular sub-domains In order to express du in terms of b_B and b_S we need to compute the determinant of the Jacobian J of the functions u_{yB} and u_{zB} with respect to the parameters b_B and b_S. We obtain:

$$J = -(1/\sin\eta)(1/H^4).$$

Note that for $\eta \neq 0$ this expression does not diverge for any real value of a_B and a_S. From (16) we have $du = |JH| db_B \wedge db_S$, thus

$$du = |(1/\sin\eta)(1/H^3)| db_B \wedge db_S.$$

Triangular sub-domains. We express H and the components of u_B in the B reference system in terms of a_B and c_B by using (15). We obtain

$$H = \frac{M}{(1 + e c_B)},$$

where

$$M = \sqrt{(1 + e c_B)^2 (1 + a_B^2) + (c_B - e)^2}. \tag{17}$$

The determinant J of the Jacobian of the functions u_{yB} and u_{zB} with respect to parameters a_B and c_B is:

$$J = \frac{-(1 + e^2)(1 + e c_B)^2}{M^4},$$

so

$$du = |JH| dc_B \wedge da_B = \frac{(1 + e^2)(1 + e c_B)}{M^3} dc_B \wedge da_B.$$

Notice that M is the square root of a sum of squares, and it cannot be zero for any real value of e, c_B and a_B.

5.7 The Projection Transformation

Lemma 2 states that the area function $A_{12}(u)$ is a polynomial in the coordinates of the vertices $Vert(u)$ in local coordinates. In order to complete our description we need to give the transformation matrix which represents the projection operation in direction u. It is sufficient to give the matrix in the coordinates u_B. Denoting with the subscript P the local reference frame on $P(u)$ having u as k_P-axis:

$$\begin{pmatrix} i_P \\ j_P \\ k_P \end{pmatrix} = \begin{pmatrix} u_{Bz}u_{Bx}/G & u_{By}u_{Bz}/G & -G \\ -u_{yB}/G & u_{xB}/G & 0 \\ u_{xB} & u_{yB} & u_{zB} \end{pmatrix} \begin{pmatrix} i_B \\ j_B \\ k_B \end{pmatrix} \tag{18}$$

Where $G = \sqrt{u_{xB}^2 + u_{yB}^2}$. In terms of coordinates:

$$\begin{pmatrix} x_P \\ y_P \\ z_P \end{pmatrix} = \begin{pmatrix} u_{Bz}u_{Bx}/G & -u_{By}/G & u_{xB} \\ u_{yB}u_{Bz}/G & u_{xB}/G & u_{yB} \\ -G & 0 & u_{zB} \end{pmatrix} \begin{pmatrix} x_B \\ y_B \\ z_B \end{pmatrix} \tag{19}$$

The local coordinates of the projection are given by x_P and y_P. Using previously derived equalities we can express the components of u_B in terms of the local spherical coordinates (α_j, β_j) for sub-domain D_j.

6 Proof of Lemma 1 for HOFF Integrals

The functions and sub-domains described in in Section 5 satisfy by construction properties (1) and (2) of definition 1. In order to prove that properties (3) and (4) are also satisfied we need to consider two subcases. The kernel of the integral is:

$$f_j(u) = A_{12}(u)$$

Thus restrictions of f_j are analytic provided that restrictions of $A_{12}(u)$ are analytic. These conditions are equivalent to requiring that: $H \neq 0$, $G \neq 0$, and $u \cdot w_F \neq 0$ for any forbidden vector F associated with R_j. Moreover the condition $H \neq 0$ also ensures that restrictions of the function $g(u)$ associated with the differential measure of direction are analytic.

6.0.1 Coordinates (b_B, b_S)

Recalling the coordinates in Subsection 5.5 for quadrangular sub-domains we have that $H \neq 0$, corresponds to the equation: $1 + a_B^2 + b_B^2 = 0$. Equivalently: $1 + (\frac{1}{\sin\eta}(b_S - (\cos\eta) b_B))^2 + b_B^2 = 0$.

For a fixed value $\bar{b_B}$ we obtain two conjugated imaginary poles in the complex b_S-plane. We obtain similarly conjugated imaginary poles when we fix the $\bar{b_S}$.

Condition $G \neq 0$ corresponds to finding zeroes of $b_B^2 + 1 = 0$, thus for any fixed value $\bar{b_S}$ we have poles at $\pm i$ on the complex b_B-plane.

Note that the various transformations introduce only imaginary poles, not real ones. Let w be a forbidden vector w_F. We show that the condition $u \cdot w = 0$ introduces only real poles. Indeed $u \cdot w = 0$ if and only if

$w_x + b_B w_y + (\frac{1}{\sin\eta}(b_S - (\cos\eta)b_B))w_z = 0$. Fixing either of the variables gives a linear equation in the other variable that has one real solution. Since in the range of the closed domain D_j the function f_j is bounded, then there is a rectangle $Rect(\bar{b_S})$ in the complex b_B-plane that strictly contains the real range I_{b_S} and none of the real or complex poles. Similarly, there is a rectangle $Rect(\bar{b_B})$ in the complex b_S-plane that strictly contains the real range I_{b_B} and none of the real or complex poles. This proves properties (3) and (4) of definition 1 for rectangular sub-domains.

6.0.2 Coordinates (a_B, c_B)

Recalling the coordinates in Subsection 5.5 for triangular sub-domains we have that $H = 0$ if and only if $M = 0$. From formula (17) we see that for a fixed value $\bar{c_B}$ we have either no roots or complex conjugated roots. For a fixed $\bar{a_B}$ equation $M = 0$ cannot have real roots in c_B. It is easy to see this by contradiction. If we suppose the existence of a real root we would have that the sum of two non negative numbers is null, which is possible only if both $(1 + ec_B)$ and $(c_B - e)$ are simultaneously zero. This is not possible for any real value of e. Thus also in this case we have two complex conjugated roots. Condition $G = 0$ is rewritten as $1 + (\frac{c_B - e}{1 + ec_B})^2 = 0$. Again, by contradiction it is easy to see that this equation cannot have real roots.

Let w be either a forbidden vector w_F. We show that the condition $u \cdot w = 0$ introduced only real poles. Indeed $u \cdot w = 0$ iff

$$(1 + ec_B)(w_x + a_B w_z) + (c_B - e)w_y = 0$$

Again this is a bilinear equation, therefore any restriction is a linear equation giving one real root. As observed before these real roots cannot be incident to the real closed domains I_{c_B} or I_{a_B}, therefore we have rectangles $Rect(\bar{c_B})$ and $Rect(\bar{a_B})$ as defined in definition 1. This proves properties (3) and (4) of definition 1 for triangular sub-domains.

7 Experimental results

The algorithm has been implemented in C and run under Linux on a Pentium 100 processor. More details on the implementation can be found in [2]. Exact reference values have been obtained from formulae in [27]. Figure 1 reports the relative error as a function of the number of Gaussian points (directions) used by the algorithm for two parallel aligned unit squares at distance $d = 2,\ 1,\ 0.5,\ 0.1$. Figure 2 reports the relative error as a function of running time. Such time values should be considered indicative only since no particular effort has been done to optimize the code. Figures 3 and 4 give relative error versus number of gaussian points and time for two unit squares sharing an edge and with a $\pi/2$ dihedral angle. Both vertical and horizontal scales are logarithmic. The graphs confirms qualitatively the exponential bound of Theorem (1).

Already for about 20 directions we have a relative error in the range 10^{-2}–10^{-3}, for about 1000 directions we can reach a precision $10^{-14} - 10^{-15}$ within twos seconds of computing time.

Results for parallel unit squares at distance 1 reported in [4, page 97] for the hemicube, Malley's algorithm and Monte Carlo integration achieve relative (maximum) error order of 10^{-2} using about 4000 points. For orthogonal unit squares results in [4, page 102] shows similar performance for Malley's method, while Monte Carlo integration and the hemicube have slower convergence.

Note that differences in computing FFs with respect to HOFFs and RFs impact only on the exact analytic calculation of the inner integral, the numerical solution of the external integral (i.e. the determination of number and position of Gaussian direction) does not depend on the basis function therefore figures 1 and 3 are indicative of performance also for HOFF and RF calculations.

The algorithm appears to be robust with respect to the relative position of the input polygons and capable of a large spectrum of time/precision trade offs. A peak precision of $10^{-14} - 10^{-15}$ is reached within a few seconds, while a standard precision of $10^{-2} - 10^{-3}$ is achieved very early in the computation process.

8 Conclusions

In this paper we have shown a method for the computation of higher order form factors between two triangles in absence of occlusions. Our algorithm can impact on practice of radiosity by making more appealing the use of larger elements together with smooth basis functions to simulate well soft shadows. Moreover the large range of error versus computing time trade-offs makes our algorithm ideal for load balancing in parallel radiosity software.

I wish to acknowledge Paolo Bientinesi's help with the experimental work.

References

[1] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 75–84. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[2] P. Bientinesi and M. Pellegrini. Electrostatic fields without singularities: implementation and experiemtns. Technical Report TR IMC B4-97-16, Istituto di Matematica Computazionale del CNR, Via Santa Maria 46, Pisa, Italy, November 1997.

[3] 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.

[4] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, San Diego, CA, 1993.

[5] P. J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, New York, 1984. Second edition.

[6] 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.

[7] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice, Second Edition*. Addison-Wesley, Reading, Massachusetts, 1990. Overview of research to date.

[8] R. Gerhbein. An adaptive method for computing irradiance coefficients of Galerkin radiosity systems. Technical Report TR-485-95, Princeton University, Dept. of Computer Science, 1995.

[9] R. Gershbein. A study of integration methods for couplings of galerkin radiosity systems. In *Eurographics Rendering Workshop 1995*. Eurographics, jun 1995.

[10] 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.

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

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

[13] P.S. Heckbert. Finite element methods for radiosity. Global illuminationcorse course notes Siggraph 93. Avalable at www.cs.cmu.edu/ ph/.

[14] P.S. Heckbert. Introduction to finite element methods. Global illumination course notes Siggraph 93. Avalable at www.cs.cmu.edu/ ph/.

[15] 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.

[16] A. Keller. *Quasi-Monte Carlo methods for photorealistic image synthesis*. PhD thesis, Department of Computer Science, University of Kaiserslauten, Germany, 1997.

[17] L. Neumann. Monte Carlo radiosity. *Computing*, 55:23–42, 1995.

[18] M. Pellegrini. Monte carlo approximation of form factors with error bounded a priori. *Discrete & Computational Geometry*, 17:319–337, 1997.

[19] M. Pellegrini. Electrostatic fields without singularities: Theory, algorithms and error analysis. *Journal of the ACM*, 45(6), November 1998. In print. Available at http://www.imc.pi.cnr.it.

[20] M. Pellegrini. Rendering equations revisited: how to avoid explicit visibility computations. In *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, pages 725–733, January 1999. Available at http://www.imc.pi.cnr.it/ marcop.

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

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

[23] Peter Schroder. Numerical integration for radiosity in the presence of singularities. In *Fourth Eurographics Workshop on Rendering*, pages 177–184, Paris, France, June 1993.

[24] Peter Schroder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. Wavelet projections for radiosity. *Computer Graphics Forum*, 13(2):141–151, June 1994.

[25] Peter Schroder and Pat Hanrahan. Wavelet methods for radiance computations. In *Fifth Eurographics Workshop on Rendering*, pages 303–311, Darmstadt, Germany, June 1994.

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

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

[28] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, 1994. excellent coverage of radiosity and global illumination algorithms.

[29] C. Soler and F. Sillion. Fast calculation of soft shadow textures using convolution. To appear in Siggraph 98 Conference Proceedings, 1998.

[30] L. Szirmay-Kalos, T. Foris, L. Neumann, and B. Csebfalvi. An analysis of Quasi-Monte Carlo integration applied to the transillumination radiosity method. *Computer Graphics Forum*, 16(3):271–281, 1997.

[31] Seth Teller and Pat Hanrahan. Global visibility algorithms for illumination computations. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 239–246, 1993.

[32] 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.

[33] Andrew J. Willmott and Paul S. Heckbert. An empirical comparison of progressive and wavelet radiosity. In Julie Dorsey and Philipp Slusallek, editors, *Eurographics Rendering Workshop 1997*, pages 175–186, New York City, NY, June 1997. Eurographics, Springer Wein. ISBN 3-211-83001-4.

[34] 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.

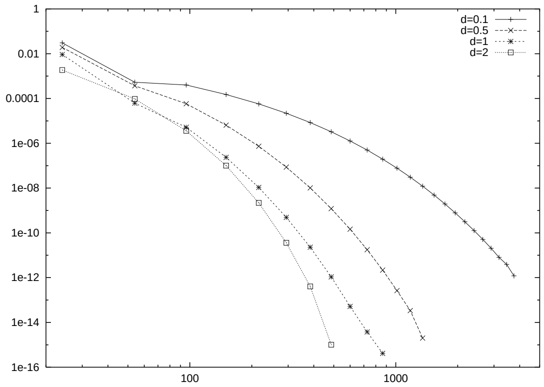

Figure 1: Parallel squares with distance d: relative error/number of directions

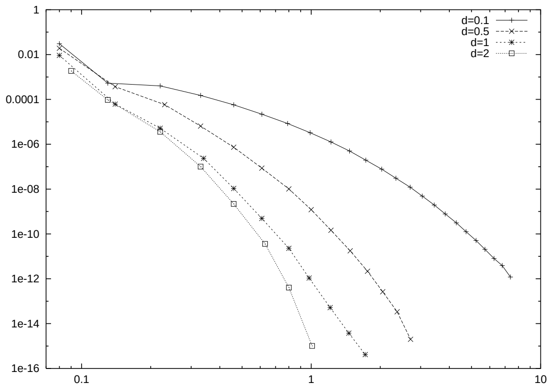

Figure 2: Parallel squares with distance d: relative error/seconds of computation

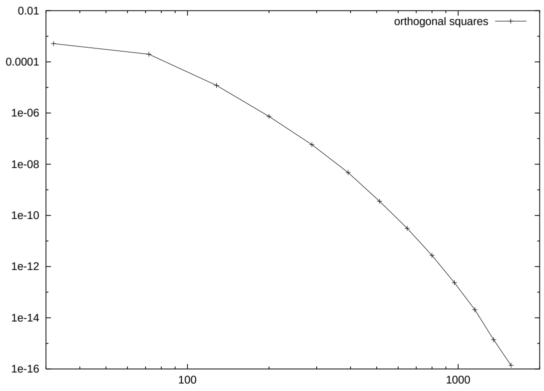

Figure 3: Orthogonal squares with a common edge: relative error/number of directions

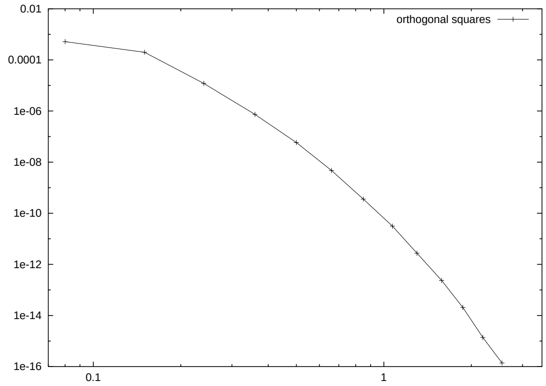

Figure 4: Orthogonal squares with a common edge: relative error/seconds of computation

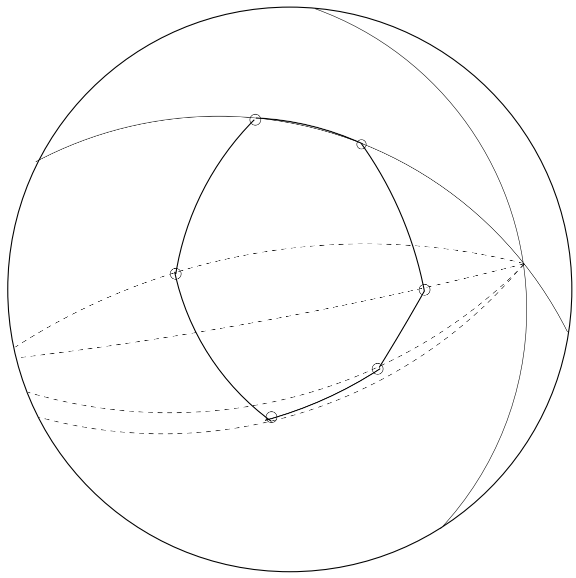

Figure 5: Slicing of a spherical polygon into sub-regions