Exact Projections onto the Hemisphere

W. Stürzlinger

GUP Linz
Johannes Kepler University Linz
Altenbergerstraße 69, A-4040 Linz, Austria/Europe
stuerzlinger@gup.uni-linz.ac.at
Tel.: +43 732 2468 9228
Fax: +43 732 2468 9496

Abstract

Recent approaches to realistic image synthesis have split the rendering process into two passes. The first pass calculates an approximate global illumination solution, the second generates an image of high quality from a given view point using the illumination solution obtained in the first pass.

This paper discusses a new method for projecting polygons onto the hemisphere which is the central operation performed for each pixel during the local pass. The method calculates an exact hemispherical projection of all polygons considering occlusion. The chosen representation of the projected polygons facilitates the fast and accurate computation of formfactors with regard to the projection center.

The new hemispherical projection method is used to obtain an exact local pass solution.

1 Introduction

A frequently encountered problem of realistic image synthesis is how to accurately compute the light reaching a surface as it determines the appearence of the surface to the viewer. How much light leaves the surface is governed by physical laws and is determined by the incoming light and the material properties of the surface.

The reflected radiance (energy per unit area) influences the illumination of other surfaces and vice versa. To capture the effects of multiple reflections global illumination methods simulate the process of radiance distribution. However it is not feasible to compute illumination for the infinite number of all points of an environment. Therefore, the problem is simplified by partitioning all surfaces into planar convex polygons. The illumination of the surface polygons is approximated by setting up an equation system which describes the exchange of radiance between them and solving it.

An alternative is to use stochastic algorithms which follow random photon rays from the light-sources. If a surface is encountered either the ray is reflected further into the scene or the energy is stored with the polygon. An overview of global illumination methods is given in [CW93,SP94].

For an immediate picture of the environment all polygons can be rendered onto the image plane, but the chosen polygon size limits the quality of the approximation and leads to image artifacts in the final picture. More correct images are obtained through a so called local pass. First the surface point visible through each pixel is calculated, then the light leaving the surface at this point is determined by calculating the incident radiance (irradiance) using the global illumination solution and taking the material properties into account.

For simplicity only the diffuse case is discussed and constant radiance is assumed for all polygons although these restrictions will be removed in chapter 3.2.

1.1 Irradiance

Assume that a solution for all radiance values L for polygons of a scene has been precalculated by an approximate global illumination simulation. The irradiance for a point P is then given by

$$H_P = \int_{\Omega} L(\vec{\omega}) \cos\Theta d\vec{\omega} \qquad (1)$$

which integrates the radiance L coming from all possible directions $\vec{\omega}$ on the hemisphere Ω weighted by the cosine of the angle θ between $\vec{\omega}$ and the normal vector of P’s surface. The radiance $L(\vec{\omega})$ depends on the surface visible in the direction $\vec{\omega}$. The main contribution of this paper is a new way to accurately solve the integral over the hemisphere.

For diffuse polygonal scenes equation (1) can be rewritten as: (see also figure 1)

$$H_P = \sum_i L_{A_i} \int_{x \in A_i} \frac{\cos\Theta_{Px} \cos\Theta'_{Px}}{\pi |P - x|^2} V(x, P)\, dx \qquad (2)$$

where the visibility function V(x,P) equals one if x and P are mutually visible and zero otherwise.

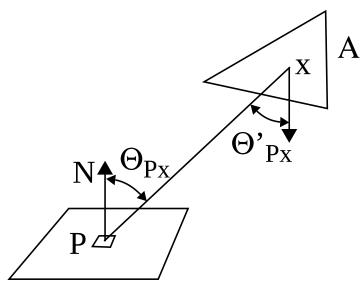

Figure 1: Geometry for local illumination

The integral in equation (2) is alsow known as the formfactor F_{PA} between point P and polygon A which simplifies the equation to:

$$H_P = \sum_i L_{A_i} F_{PA_i} \qquad (3)$$

For unoccluded polygons the formfactor for a polygon can be calculated by solving the contour integral (see e.g. [BRW89]):

$$F_{PA} = \frac{1}{2\pi} \sum_{a \in A} N \cdot \Gamma_a \qquad (4)$$

where a are the edges of the polygon A and N is the surface normal at P. Γ_a is the normal to the plane defined by P and the edge a with magnitude equal to the angle γ_a (see figure 2).

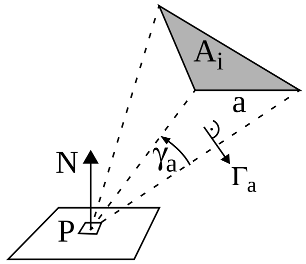

Figure 2: Formfactor from a point to a polygon

For an alternative way formulation for computing formfactors using projected great circles see [Bi92].

The following section describes a new method of projection polygons onto the hemisphere taking occlusion into account. To compute unoccluded formfactors equation (4) is used.

2 The Hemispherical Projection

2.1 Previous Work

First approximations to a projection onto the hemisphere were computed using a hemicube [CG85]. The polygons are projected onto the five faces of a half cube. Other approaches use other discretizations to the hemisphere's surface [Sp91,BP91], the projection onto a single plane [SP89], or monte carlo ray tracing [Ma88] to compute formfactors.

The above methods have in common that they do not deliver exact results for arbitrary polygonal environments with occlusion.

2.2 Exact Projection onto the Hemisphere

First the projection of a polygon A with vertices v_j onto the unit hemisphere defined by a projection center P and its normal vector N is desribed.

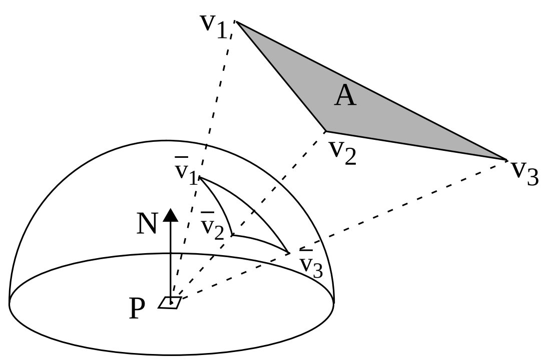

Figure 3: Projection of polygon onto hemisphere

Simply normalizing the vector $\vec{Pv_j}$ for each vertex of A gives the projected vertices $\bar{v}_j$. The projection of the polygon edges are segments of great circles. The projected vertices $\bar{v}_1$ and $\bar{v}_2$ and the line between them all lie in the plane defined by P, v_1 and v_2. Viewed from the projection center P the line and the circle segment coincide.

A projected polygon covers a part of hemisphere limited by great circles and is uniquely defined by its projected vertices $\bar{v}_j$.

Testing beforehand if the polygon A faces the projection center P removes backfacing polygons. Clipping the polygon A at the plane defined by P and N assures that all projected vertices lie on half of the unit sphere at P therefore no coordinate transformation is necessary to align the normal vector N with an axis.

2.3 Intersecting Projections

Consider the intersecting projections of two polygons $\bar{A}$ and $\bar{A}'$. On the hemisphere their edges are segments of great circles. Instead of intersecting circles the intersection of the projections can be computed easily as follows.

Two edges intersect if the vertices $\bar{v}_j$ and $\bar{v}_{j+1}$ are separated by the plane defined by the projection center P and the two vertices $\bar{v}_k'$ and $\bar{v}_k'_{+1}$ and vice versa. The intersection point $\bar{i}$ is computed by intersecting one of the planes with the line connecting the other two vertices in three dimensional space and projecting it onto the hemisphere by normalizing its vector (see figure 4).

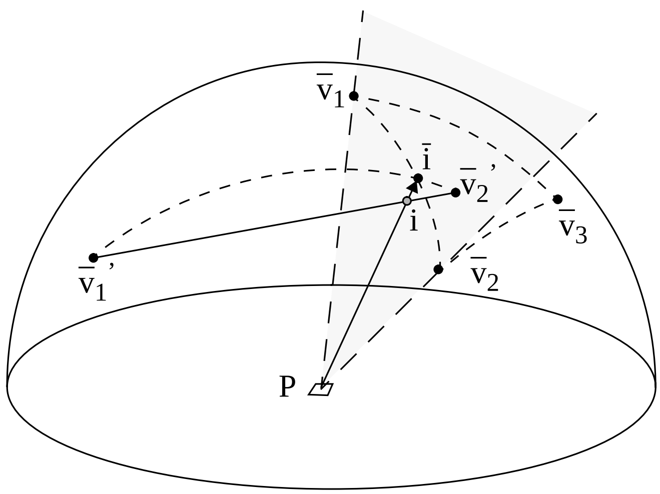

Figure 4: Intersection of two projected edges

Integrating this edge intersection method into a standard two dimensional polygon intersection algorithm allows to accurately intersect projected polygons on the hemisphere. Any data structure for two dimensional polygonal partitions (e.g. winged-edge,...) can be used to store the partition of the hemisphere. Note that all points are three dimensional and all tests and intersection calculations are done in three dimensional space.

2.4 Visible Surface Determination

With each projected vertex the distance from the projection center P to the original vertex v_j is stored as $dist(\bar{v}_j)$ and for each projected polygon a reference to the original polygon is maintained as well.

Now let A and A' be two polygons the projections $\bar{A}$ and $\bar{A}'$ of which intersect on the hemisphere, and assume A' is further away from the projection center P than A. Then $\bar{A}'$ has to be clipped to the portion outside of $\bar{A}$ using the intersection method described in chapter 2.3. If the original polygons do not actually intersect in three dimensional space $dist(\bar{v}_j)$ is always smaller than $dist(\bar{v}_j')$

in the region of overlap between $\overline{A}$ and $\overline{A}$'. A similar case holds if A' lies completely before A.

If the distances *dist*($\bar{v}_j$) and *dist*($\bar{v}_j$') indicate that not all v_j lie before all v_j' then the polygons intersect in three dimensional space. This case can be handled by calculating the plane defined by P and the three dimensional intersection line and using this plane to partition the projected polygons.

2.5 Calculating Irradiance

The formfactor $F_{P\overline{A}}$ from a point P to a projected polygon $\overline{A}$ is computable by equation (4) as it is based on angles between vectors and does not assume a planar polygon. Weighting the radiance of each projected polygon by its formfactor and summing over all projected polygons gives the irradiance H_P (see equation (3)) incident at point P.

The calculated formfactors are exact whereas the accuracy of the irradiance H_P depends on the error of the global illumination solution.

3 The Local Pass

3.1 The Local Illumination Model

The local illumination model describes the relationship of the irradiance E of a point P (the "incoming" light) and its outgoing radiance L.

The (outgoing) radiance of a point P on a diffuse surface equals its irradiance multiplied by the reflectivity of the surface plus its emission:

$$L_P = E_P + \frac{\rho_P}{\pi} H_P \qquad (5)$$

where E_P is the emission and ρ_P is the reflectivity of P's surface.

3.2 The Local Pass

The local pass avoids the artefacts of the discretized global illumination solution. These artifacts are due to an interpolation of radiance values at inappropriate places or over unacceptably large areas as neither the final view point nor the image resolution are before the placement of sampling points. Based on a coarse solution of the global illumination problem a high quality picture is generated by finding the point P visible at each pixel and obtaining the colour of this pixel by computing the irradiance of P and using equation (5) to compute its radiance. Previous work employed either the hemicube [Re92], raytracing (e.g. [WEH89]), or stochastic methods (e.g. [Ru88,Wa94,Sh91]) to compute approximate solutions to the irradiance.

The new hemisphere projection algorithm allows to compute the exact irradiance and radiance of an arbitrary surface point in diffuse polygonal scenes. The local illumination model is evaluated accurately to within the precision of the global illumination simulation.

3.3 Generalization to Arbitrary Surfaces

For surfaces with general material properties the outgoing radiance of P in the view direction $\vec{\omega}'$ is given by:

$$L_P(\vec{\omega}') = E_P(\vec{\omega}') + \int_{\Omega} \rho_{bd}(P, \vec{\omega}, \vec{\omega}')\, L(\vec{\omega}) \cos\Theta\, d\vec{\omega} \qquad (6)$$

where the irradiance from direction $\vec{\omega}$ is weighted by the bidirectional reflection distribution function (BRDF) ρ_{bd} of P's surface in the viewing direction $\vec{\omega}'$ and by the cosine of the angle θ between $\vec{\omega}$ and the normal vector of P's surface. For arbitrary scenes and general BRDF's this integral can be evaluated only numerically due to the dicontinuities in the integrand.

The above integral can be rewritten:

$$L_P(\vec{\omega}') = E_P(\vec{\omega}') + \sum_i \int_{\vec{\omega} \in \bar{A}_i} \rho_{bd}(P, \vec{\omega}, \vec{\omega}') L_i(\vec{\omega}) \cos\Theta d\vec{\omega} \tag{7}$$

where $L_i(\vec{\omega})$ is the radiance of the polygon A_i in the direction $\vec{\omega}$.

Now the integral has to be evaluated over polygonal regions which can be simplified further by partitioning the regions into triangles if necessary. Assuming a smooth BRDF the integrand will be a smooth function also, and therefore, this integral is significantly easier to compute than equation (6). Several optimizations are possible, e.g. the number of evaluations of the integrand can be adapted to the solid angle subtended by the projected polygon $\bar{A}_i$.

3.4 Optimizing the Local Pass

The complexity of the local pass is proportional to the number of pixels and the number of polygons as all polygons have to be projected onto the hemisphere for each pixel. One way to reduce the cost is to evaluate the local illumination integral not for all pixels and use some kind of interpolation scheme for the others. This is not dicussed in this article.

The other option is to reduce the cost for the hemispherical projection. Assume that all polygons are stored in a space hierarchy. Traversing the hierarchy front to back and projecting the polygons stored in each voxel is equivalent to a depth sort of the polygons. If the region of the hemisphere corresponding to the projected silhouette of a voxel is already "covered" by polygons, the voxel needs no further consideration.

4 Implementation and results

The new hemisphere projection method was integrated into a standard diffuse radiosity program and used to compute the local pass. In the following tables the following methods are compared

- Projection onto a randomly rotated hemicube using a hardware z-buffer with two resolutions.
- A projection method based on [WEH89] but using equation (4) instead of the disk approximation for formfactor calculation. The number of rays is adapted to the distance of the polygon.
- The new hemisphere projection method using a two dimensional BSP tree to speed the intersection calculations without the optimization described in chapter 3.4.

The current implementation runs on a Silicon Graphics Indigo Workstation (R3000) with an XS24Z hardware z-buffer. For two scenes with 176 and 9504 polygons the average time per pixel and is given in seconds. For an simple accuracy comparison of the maximum deviation of the formfactor sum from one is given as well. For any hemispherical projection method the sum of formfactors in an arbitrary closed polygonal environment should equal one.

Judging from the formfactor sum the hemicube method is the most accurate, but due to aliasing effects the error in the irradiance values may be arbitrarily high.

Method	Hemicube 50x50	Hemicube 200x200	Raytraced FF	Hemisphere projection
max FF sum	0.000217	0.000014	0.12122	0.00005
Time/pixel	0.0625	0.1111	0.5694	0.6944

Table 5: Scene 1 with 176 polygons

Method	Hemicube 50x50	Hemicube 200x200	Raytraced FF	Hemisphere projection
max FF sum	0.000217	0.000014	0.4012	0.00004
Time/pixel	0.4722	0.5000	6.2431	21.5930

Table 6: Scene with 9504 polygons

5 Conclusion and Further Extensions

The presented method introduces an exact way to compute the hemispherical projection of an arbitrary polygonal scene with an arbitrary surface point as projection center. This allows to accurately compute the irradiance for surface points using the results of a global illumination solution.

The hemisphere projection facilitates the calculation of the irradiance jacobian [Ar94]. All necessary quantities can be computed from the obtained projection onto the hemisphere. Using the irradiance jacobian it should be possible to interpolate irradiance values in areas of the image where the irradiance varies slowly (see e.g. [WH92]).

Incoporating methods similar to the hierarchical z-buffer [GKM93] scheme should speed up the projection. A possible further extension is the use of error bounds to allow approximate projections.

6 Bibliography

[Ar94] J. Arvo. The Irradiance Jacobian for Partially Occluded Polyhedral Sources. Computer Graphics (SIGGRAPH 94), pp. 343-350, July 1994

[Bi92] B. Bian. Hemispherical Projection of a triangle. In Graphics Gems III, D. Kirk, Ed. Academic Press, San Diego, 1992, pp. 314-317.

[BP91] J. C. Beran-Koehn, and M. J. Pavicic. A cubic tetrahedal adaption of the hemi-cube algorithm. In Graphics Gems II, J. Arvo, Ed. Academic Press, San Diego, 1991, pp. 229-302.

[BRW89] D. R. Baum, H. E. Rushmeier, and J. M. Winget. Improving radiosity solutions through the use of analytically determined form-factors. Computer Graphics (SIGGRAPH 89), pp. 325-334, July 1989

[CG85] M. F. Cohen, and D. P. Greenberg. The hemicube: A radiosity solution for complex environments. Computer Graphics (SIGGRAPH 85), pp. 31-40, July 1985.

[CW93] M. F. Cohen, and J. R. Wallace. Radiosity and Realistic Image Synthesis. Academic

Press, 1993.

[GKM93] N. Greene, M. Kass, G. Miller. Hierarchical Z-Buffer Visibility. Computer Graphics (SIGGRAPH 93), pp. 231-238, Aug. 1993.

[Ma88] T. J. Malley. A shading method for computer generated images. Master's thesis, University of Utah, June 1988.

[Re92] Mark C. Reichert. A two-pass radiosity method driven by lights and viewer position. Master's thesis, Cornell University, Ithaca, NY, Jan. 1992.

[Ru88] H. E. Rushmeier. Realistic Image Synthesis for Scenes with Radiatively Participating Media. Ph.D. thesis, Cornell University, 1986.

[Sh91] P. Shirley. Physically Based Lighting Calculations for Computer Graphics. Ph.D. thesis, University of Illinois at Urbana-Champaing, 1991.

[Sp91] S. N. Spencer. The hemisphere radiosity method: A tale of two algorithms. In Photorealism in Computer Graphics (Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics, 1990), K. Bouatouch and C. Bouville, Eds., 1991, pp. 127-135.

[SP89] F. Sillion, and C. Puech. A general two-pass method integrating specular and diffuse reflection. Computer Graphics (SIGGRAPH 89), pp. 335-344, July 1989.

[SP94] F. X. Sillion, and C. Puech. Radiosity and Global Illumination. Morgan Kaufmann Publishers, 1994.

[Wa94] G. J. Ward. The RADIANCE Lighting Simulation and Rendering System. Computer Graphics (SIGGRAPH 94), pp. 459-472, July 1994.

[WEH89] J. R. Wallace, K. A. Elmquist, E. A. Haines. A ray tracing algorithm for progressive radiosity. Computer Graphics (SIGGRAPH 89), pp. 315-324, July 1989.

[WH92] G. Ward, and P. Heckbert. Irradiance Gradients. In Proceedings of Third Eurographics Workshop on Rendering, pp. 85-98, May 1992.