THE HEMI-CUBE
A RADIOSITY SOLUTION FOR COMPLEX ENVIRONMENTS

Michael F. Cohen and Donald P. Greenberg
Cornell University
Ithaca, N. Y. 14853

ABSTRACT

This paper presents a comprehensive method to calculate object to object diffuse reflections within complex environments containing hidden surfaces and shadows. In essence, each object in the environment is treated as a secondary light source. The method provides an accurate representation of the "diffuse" and "ambient" terms found in typical image synthesis algorithms. The phenomena of "color bleeding" from one surface to another, shading within shadow envelopes, and penumbras along shadow boundaries are accurately reproduced. Additional advantages result because computations are independent of viewer position. This allows the efficient rendering of multiple views of the same scene for dynamic sequences. Light sources can be modulated and object reflectivities can be changed, with minimal extra computation. The procedures extend the radiosity method beyond the bounds previously imposed.

CR Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.3.3 [Computer Graphics]: Picture/Image Generation

General Terms: Algorithms

Additional Keywords and Phrases: Radiosity, diffuse reflections, hidden surface, form-factors, depth buffer.

INTRODUCTION

The representation of a realistic image of both actual and imagined scenes has been the goal of artists and scholars for centuries. The invention of the camera allowed the photographer to mechanically record the light passing through a lens and focused onto a piece of film, thus producing a realistic image. Today, within the field of computer graphics, one aspect of current research has been the attempt to produce realistic images of non-existent scenes. This is accomplished by simulating the distribution of light energy given a geometric and physical description of the environment. One major difficulty has been the correct simulation of the global illumination effects.

Light leaving an object surface originates from the surface by direct emission, as from a light source, or by the reflection or transmission of incident light. The incident light on a surface can arrive directly from a light source (along a direct line of sight) or indirectly by generally complex intermediate reflections and transmissions from other surfaces within the environment. Previously these secondary light "sources" have been ignored in computer graphics image generation algorithms. The summation of these sources have usually been approximated by an added constant term, referred to as the ambient component. [9] Some aspect of the global illumination has been achieved by the addition of a specular component found in ray tracing algorithms [13]. In essence, rays are traced only in the mirror reflection or transmission directions, point sampling the environment at the specific surface intersections. Ray tracing with

cones [1] or distributed ray tracing [2] extends this procedure by gathering light from more than one point per ray, but can not accurately model the global environmental lighting effects.

The majority of surfaces in a real environment are "diffuse" reflectors, i.e., an incident beam of light is reflected or scattered in all directions within the entire hemisphere above the reflecting surface. A special case of diffuse reflection is the so-called "ideal" diffuse or "Lambertian" reflection. In this case, the incident light is reflected from a surface with equal intensity in all directions. Ideal diffuse reflection is assumed in this paper. Specular reflections, from mirror-like surfaces, which account for a much smaller proportion of the reflected light energy, are not considered.

Thermal engineers have previously developed methods to determine the exchange of radiant energy between surfaces. [10] [11] Methods have been developed to determine the energy exchange within enclosures. The application of one such method, known as the radiosity method to computer graphics, was outlined in a paper by Goral. [5]

This paper extends the use of the radiosity method, to a broader class of problems. In particular, complex environments with occluded surfaces are allowed. In addition, very efficient procedures to render an image from the radiosity data are discussed and illustrated with examples.

RADIOSITY

The radiosity method describes an equilibrium energy balance within an enclosure. The essential features are summarized here for completeness. [5] It is assumed that all emission and reflection processes are ideal diffuse. Thus, after reflection from a surface, the past history or direction of a ray is lost.

The light leaving a surface (its radiosity) consists of self-emitted light and reflected or transmitted incident light. The amount of light arriving at a surface requires a complete specification of the geometric relationships among all reflecting and transmitting surfaces, as well as the light leaving every other surface. This relationship is given by:

$$\text{Radiosity}_i = \text{Emission}_i + \text{Reflectivity}_i \int_{env} \text{Radiosity}_j \text{Form-factor}_{ij} \qquad (1)$$

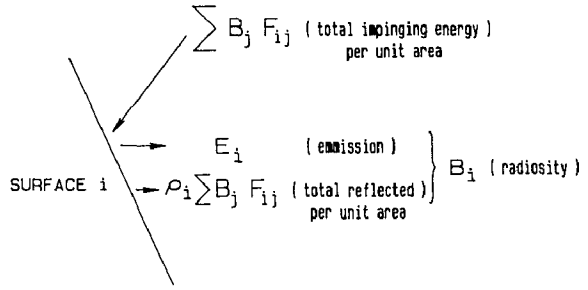
Radiosity (B): The total rate of energy leaving a surface. Sum of emitted and reflected energy. (energy/unit time/unit area)

Emission (E): The rate of energy (light) emitted from a surface. (energy/unit time/unit area)

Reflectivity (P): The fraction of incident light which is reflected back into the environment. (unitless)

Form-factor (F): The fraction of the energy leaving one surface which lands on another surface. (unitless)

This equation states that the amount of energy (or light) leaving a particular surface is equal to the self-emitted light plus the reflected light. The reflected light is equal to the light leaving every other surface multiplied by both the fraction of that light which reaches the surface in question, and the reflectivity of the receiving surface. The sum of the reflected light from a given surface plus the light emitted directly from the surface is termed its radiosity. (Figure 1)



RADIOSITY RELATIONSHIPS

Figure 1

If the environment is subdivided into discrete surface elements or "patches", for which a constant radiosity is assumed, a set of simultaneous equations can be generated to describe the interaction of light energy within the environment. [5]

These equations take the form: (2)

$$\begin{bmatrix} 1 - p_1F_{11} & -p_1F_{12} & \cdots & -p_1F_{1N} \\ -p_2F_{21} & 1-p_2F_{2N} & \cdots & -p_2F_{2N} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ -p_NF_{N1} & -p_NF_{N2} & \cdots & 1-p_NF_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \cdot \\ \cdot \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \cdot \\ \cdot \\ E_N \end{bmatrix}$$

The color of an object is determined by its reflectivity (or emission in the case of a light source) at each wavelength of the visible spectrum. The reflectivity and emission terms in the above equations are, therefore, valid for a particular wavelength or band of wavelengths. It is necessary to form and solve the above matrix for each band of interest in order to determine the full radiosity of each patch. For the current discussion, it is sufficient to state that each point or patch is assumed to have a single constant radiosity B, even though this radiosity may be a set of three or more values to represent the radiosity at the different bands of interest. It is important to note that the form-factors are solely a function of geometry and are thus independent of any color considerations.

The FORM-FACTOR

The form-factor specifies the fraction of the energy leaving one surface which lands on another. By definition the sum of all the form-factors from a particular point or patch is equal to unity. Previous derivations do not account for occluding surfaces, and have therefore not reproduced shadows and penumbras.

The geometric terms in the form-factor derivation are illustrated in Figure 2. For non-occluded environments the form-factor for one differential area to another is given by:

$$F_{dAi\ dAj} = \frac{\cos\phi_i \cos\phi_j}{\pi r^2} \qquad (3)$$

By integrating over area j , the form-factor from a finite area (or patch) to a differential area can be expressed:

$$F_{dAi\ Aj} = \int_{A_j} \frac{\cos\phi_i \cos\phi_j}{\pi r^2}\, dAi \qquad (4)$$

The form-factor between finite surfaces (patches) is defined as the area average and is thus:

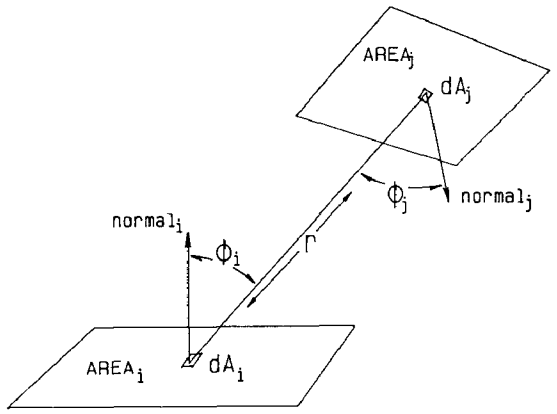
$$F_{Ai\ Aj} = \frac{1}{Ai} \int_{AiAj} \int \frac{\cos\phi_i \cos\phi_j}{\pi r^2}\, dAj\ dAi \qquad (5)$$

This expression for the form-factor does not account for the possibility of occluding objects hiding all or part of one patch from another. There is, therefore, a missing term within the integrand if hidden surfaces are to be accounted for.

$$F_{Ai\ Aj} = \frac{1}{Ai} \int_{AiAj} \int \frac{\cos\phi_i \cos\phi_j}{\pi r^2}\, HID\ dAj\ dAi \qquad (6)$$

The function (HID) takes on a value of one or zero depending on whether differential area i can "see" differential area j. The HID function has the effect of producing the projection of area j visible from differential area i. It is the solution for this double area integral (6), which must be found to solve for radiosities in any non-convex environment.

In the past this double area integral has proven difficult to solve analytically for general applications. Form-factors between specific shapes and orientations, such as parallel rectanglar plates or circular disks, have been solved and tabulated [10]. An area integral, which is a double integral itself, can be transformed via Stoke's theorem into a single contour integral which can then be evaluated [5] [11]. For non-occluded environments the double area integral can be transformed into a double contour integral. A more general approach is needed to handle complex geometries.



FORM-FACTOR GEOMETRY

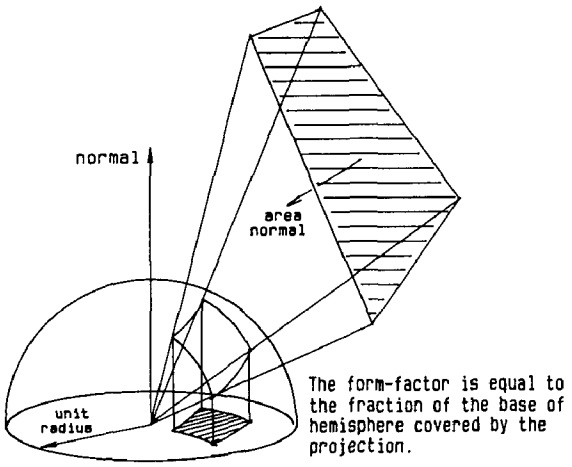
Figure 2

Numerical techniques can provide a more efficient means to compute form-factors for general complex environments. Starting from a geometric analog to the analytic derivation, a numerical method is outlined to approximate the patch to patch form-factors which includes the hidden surface effects.

If the distance between the two patches is large compared to their size, and they are not partially occluded from one another, it can be seen that the integrand of the inner integral remains almost constant. In that case the effect of the outer integral is simply a multiplication by one and finding a solution to the inner integral will provide a good approximation for the form-factor. If the patches are close together relative to their size, or there is partial occlusion, the patches can be subdivided into smaller patches and the single integral equation approximation can still be used.

The form-factor from patch to patch is approximated with the differential area to finite area equation (4) by using the center point of patch i to represent the average position of patch i. Each patch has as its "view" of the environment, the hemisphere surrounding its normal.

A geometric analog for the form-factor integral was developed by Nusselt [10] and has been used to obtain form-factors by both photography and planimetry. (Figure 3). For a finite area, the form-factor is equivalent to the fraction of the circle (which is the base of the hemisphere) covered by projecting the area onto the hemisphere and then orthographically down onto the circle.



The projection onto the hemisphere accounts for the 1/r term as well as the cosine of the angle between the normal of the projecting patch and the r vector. The orthographic projection onto the circle has the effect of multiplying by the cosine of the angle between the "r" vector and the normal of the receiving patch. The PI in the denominator accounts for the area of a circle with unit radius.

NUSSELT ANALOG

Figure 3

This hemisphere can be broken into small delta solid angles, each representing a delta form-factor. The delta form-factor is equal to the fraction of the hemisphere subtended by the delta solid angle multiplied by the cosine of the angle from the normal. The form-factor to a patch is equal to the sum of the delta form-factors covered when projecting the patch onto the hemisphere. If all the patches in the environment are projected onto the hemisphere, removing the more distant areas in the case of an overlap, the final projection and summations would provide the form-factors to all patches from the patch represented at the center of the hemisphere. This procedure would then intrinsically include the effects of hidden surfaces.

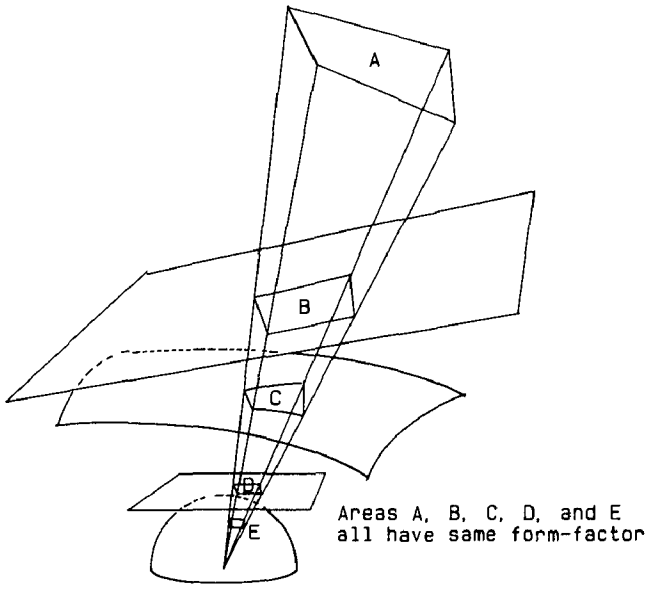
In order to perform the summation of delta form-factors for each patch one would like a convenient means for discretizing the surface of the hemisphere. An evaluation can then be made as to which patch projects onto that discrete area. Difficulty in creating equal sized elements on a sphere as well as creating a set of linear coordinates to uniquely describe locations on its surface makes this approach impractical.

THE HEMI-CUBE

An efficient algorithm to compute form factors in general complex environments is described below.

From the definition of the form-factor it can be seen that any two patches in the environment, which when projected onto the hemisphere occupy the same area and location, will have the same form-factor value. This is also true for projections onto any other surrounding surface. (Figure 4)

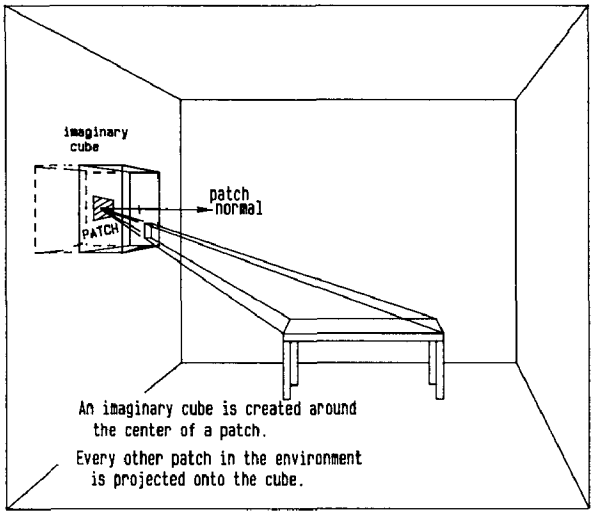
Instead of projecting onto a sphere, an imaginary cube is constructed around the center of the receiving patch. (Figure 5) The environment is transformed to set the patch's center at the origin with the patch's normal coinciding with the positive Z axis. In this orientation the hemisphere described above is replaced by the upper half of the surface of the cube, the lower half being below the "horizon" of the patch. One full face, facing in the Z direction and four half-faces, facing in the +X -X +Y and -Y directions replace the hemisphere. These faces are divided into square "pixels" at a given resolution, generally between 50 x 50 and 100 x 100, and the environment is then projected onto the five planar surfaces. (Figure 6) Each full face of the cube covers exactly a 90 degree frustum as viewed from the center of the cube. This creates clipping planes of Z = X , Z = -X , Z = Y, and Z = -Y , allowing for simple comparisons to determine which side of a clipping plane any point lies. Every other patch in the environment is clipped to the frustum using a Sutherland-Hodgman type polygon clipper [8] streamlined to handle 90 degree frustums.



AREAS WITH IDENTICAL FORM-FACTOR

Figure 4

If two patches project onto the same pixel on the cube, a depth determination is made as to which patch is "seen" in that particular direction by comparing distances to each patch and selecting the nearer one. This depth buffer type hidden surface algorithm is well known in computer graphics. [3] However, rather than maintaining intensity information, an item buffer is maintained of which patch is seen at each pixel on the cube. [12]



PROJECTION OF THE ENVIRONMENT ONTO THE HEMI-CUBE

Figure 5

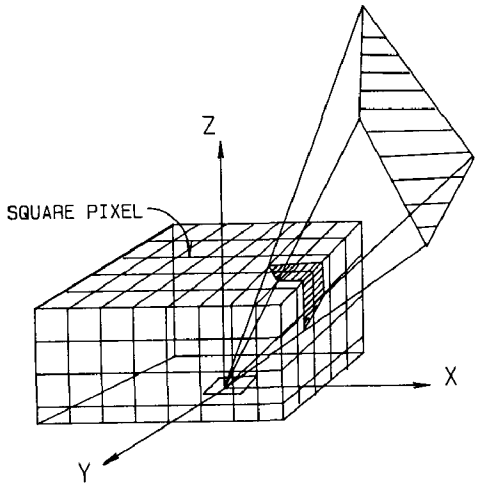
After determining which patch(j) is visible at each pixel on the hemi-cube, a summation of the delta form-factors for each pixel occupied by patch(j) determines the form-factor from patch (i) at the center of the cube to patch(j). This summation is performed for each patch(j) and a complete row of N form-factors is found.

At this point the hemi-cube is positioned around the center of another patch and the process is repeated for each patch in the environment. The result is a complete set of form-factors for complex environments containing occluded surfaces.

$$F_{ij} = \sum_{q=1}^{R} \Delta F_q \qquad (7)$$

ΔF_q = Delta form-factor associated with $pixel_q$ on hemi-cube

R = Number of hemi-cube pixels covered by projection of patch onto the hemi-cube



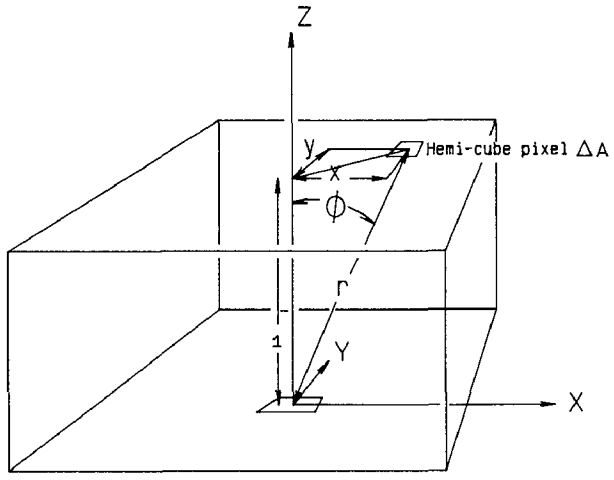
THE HEMI-CUBE

Figure 6

DELTA FORM FACTORS

The contribution of each pixel on the cube's surface to the form-factor value varies and is dependent on the pixel location and orientation. (Figure 7)

A specific delta form-factor value for each pixel on the cube is found from equation (3) for the differential area to differential area form-factor and stored in a lookup table. This table need only contain values for one eighth of the top face and one quarter of one side face due to symmetry.



TOP OF HEMI-CUBE

$$r = \sqrt{x^2 + y^2 + 1}$$

$$\cos\phi_i = \cos\phi_j$$

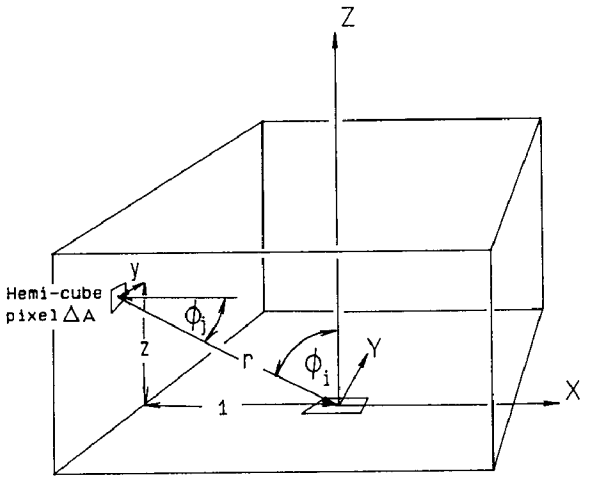
$$\cos\phi = \frac{1}{\sqrt{x^2 + y^2 + 1}}$$

$$\Delta\text{Form-factor} = \frac{\cos\phi_i \cos\phi_j}{\pi r^2}\Delta A$$

$$= \frac{1}{\pi (x^2 + y^2 + 1)^2}\Delta A$$

DERIVATION OF DELTA FORM-FACTORS

Figure 7a



SIDE OF HEMI-CUBE

$$r = \sqrt{y^2 + z^2 + 1}$$

$$\cos \phi_i = \frac{z}{\sqrt{y^2 + z^2 + 1}}$$

$$\cos \phi_j = \frac{1}{\sqrt{y^2 + z^2 + 1}}$$

$$\Delta \text{Form-factor} = \frac{\cos \phi_i \cos \phi_j}{\pi r^2} \Delta A$$

$$= \frac{z}{\pi (y^2 + z^2 + 1)^2} \Delta A$$

DERIVATION OF DELTA FORM-FACTORS

Figure 7b

THE ENERGY BALANCE SOLUTION

The solution for the series of simultaneous equations can be performed with any standard equation solver. However, a Gauss-Siedel iterative approach has a number of advantages. [6] The matrix is well suited to this technique due to the fact that it is diagonally dominant (the sum of the absolute values of each row is less than the main diagonal term). This is always true since, by definition, the sum of any row of form-factors is equal to unity. In the matrix to be solved, each form factor term is multiplied by its surface reflectivity, which is also less than unity. Thus the summation of the absolute values of all terms in any row exclusive of the main diagonal term is also less than one. The main diagonal term is equal to one minus its own form-factor. Since any polygonal (or convex) patch cannot see itself, its own form-factor is equal to zero. Therefore, the main diagonal term is always equal to one, the matrix is strictly diagonally dominant, and guaranteed to converge rapidly to a solution.

An initial guess for the radiosities, which must be supplied for the first iteration, is simply the emission of each patch (only the primary light sources have any initial radiosity). During each iteration each radiosity is solved for, using the previously found values of the other radiosities. Iterations continue until no radiosity value changes by more than a preselected small percentage. The iterative process converges very rapidly, generally in six to eight iterations, and a solution is found in a fraction of the time needed for standard elimination techniques.

Additional computational savings are made by compressing the null entries from the matrix. Zero valued form-factors will occur when one patch cannot "see" another due to occluding surfaces, if they belong to the same polygon, or face away from each other. Zeros will also occur in the matrix if reflectivities are equal to zero.

The matrix is formed and solved for the radiosities from the computed form-factors for each color band of interest. This is generally performed for three channels (red, green, blue) but could be done on a wavelength basis if desired.

RENDERING

To render an image the discretized radiosity information is used to create a continuous shading across a given surface (or polygon). A number of shading schemes have been devised in the past [5], however, most of these take place in image space and are axis dependent. An object space smoothing algorithm is desired in order to be able to render sequences of different views such that the same point in space would retain the same intensity values.

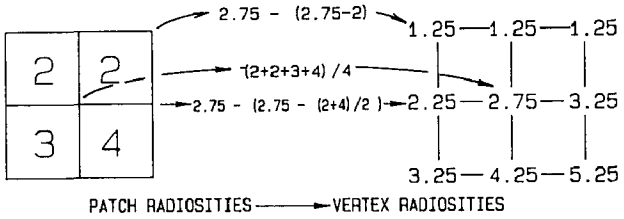
The adopted method uses a bilinear interpolation within each patch. This bilinear variation of radiosities insures first order continuity at patch edges. In order to perform the interpolation, radiosity values must be transferred from the patches themselves to each vertex of the patches. (Figure 8a)

For each polygon, the radiosities of patch vertices which are interior to the polygon containing the patch are computed as the average of radiosities of the surrounding patches. Exterior vertex radiosities are extrapolated values from the adjacent interior vertex radiosities through an average of the adjacent patch radiosities. Finally, the resulting vertex radiosities are used for the bilinear interpolation within each patch. (Figure 8b)

The radiosity solution is independent of view position and direction. To render an image on a raster display device, the eye position as well as the viewing direction and frustum angle must be specified. A transformation matrix sets the eye and view direction within the environment. A depth-buffer/item-buffer algorithm determines which patch is seen at each pixel of the screen.

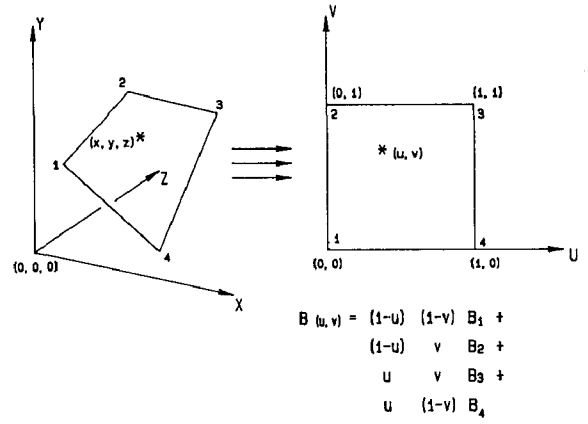
Finding the intersection between a line from the eye through a pixel, and the plane of the patch which occupies the pixel provides a location within the patch. In order to perform the bilinear interpolation within the patch, this x,y,z position is converted to parametric (u,v) coordinates within the patch.

With this information bilinear interpolation of the three radiosity values are made and the pixel displayed.



VERTEX RADIOSITIES

Figure 8a



BILINEAR INTERPOLATION OF PIXEL RADIOSITY FROM VERTEX RADIOSITIES

Figure 8b

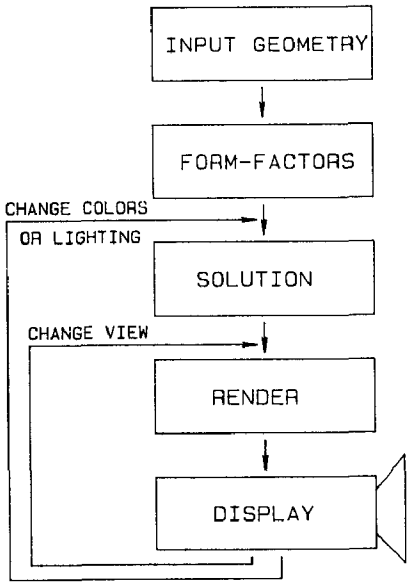
PROGRAM SUMMARY (Figure 9)

Input: The environment geometry is input from a file containing polygon descriptions and the associated vertex coordinates. Associated with each polygon are reflectivity and emission values for each color band and a parameter for subdividing the polygon into patches.

Form-factors: After the patches are defined, the hemi-cube is set around the center of each patch and a row of form-factors is determined from that patch to all others through their projections onto the cube. A file containing the matrix of form-factors is created.

Radiosity solution: For each color band a matrix is constructed using the form-factors and the appropriate set of reflectivities. The corresponding emission values are then used to solve for patch radiosities. A file containing the radiosities for each color band is written.

Rendering: An eye position, view direction, and frustum angle are specified from which an image is rendered. For each pixel, the location of the ray-patch intersection is computed. The color(s) are found through a bilinear interpolation of the vertex color values and then displayed.



PROGRAM FLOW

Figure 9

SIMULATION VS REALITY

An experimental setup was devised to test the accuracy of computer simulations. A physical model of the simple environment simulated in figure 10 was constructed of wood and cardboard. The light and painted surfaces were made as diffuse as possible. Radiometric measurements were made at locations on the model and compared with the calculated values. A visual side-by-side comparison was made by a group of experimental subjects. Great lengths were taken to control the test conditions for both methods of comparisons. The results from both tests provide conclusive validation of the radiosity method. Details and illustrations of the experimental setup and the results are contained in an article by Meyer et. al. [7].

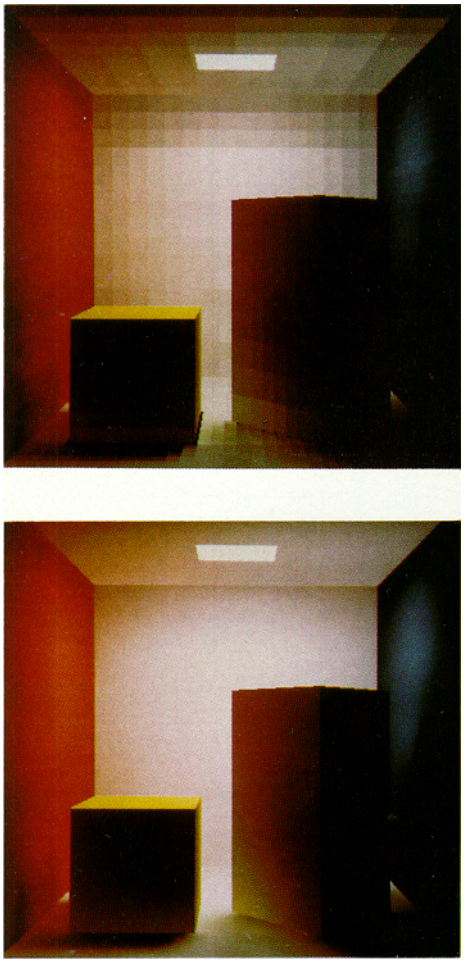


Figure 10

The environment simulated above, although simple in geometry displays a variety of shading characteristics. The patch radiosities are displayed on the top as constantly shaded and on the bottom after the bilinear interpolation for pixel radiosities.

Note:

(1) The distinct color bleeding onto the white and yellow boxes.

(2) The red shadow to the left of the yellow box. The majority of the light which reaches the shaded area arrives indirectly from the red wall, thus the red shadow.

(3) The variation of light intensity on the walls due to the diffuse light source.

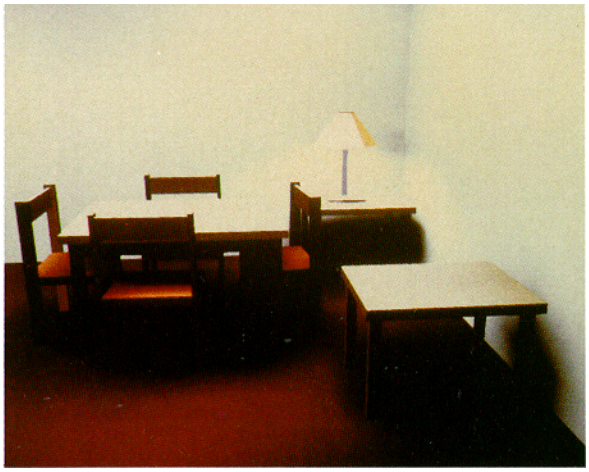
No. of Patches: 2370

Form-factor computation: 337 Min.

Solution of matrices: 18 Min.

Rendering and display: 14 Min.

(A)



(B)

(C)

Figure 11

*All computation was performed on a VAX 11/780. The images were displayed on a Grinnell 512 x 480 frame buffer.

The images of figure 11 demonstrate the computational savings achieved when rendering multiple images of the same environment.

No. of patches: 1740	Image A	Image B	Image C
Form-factor comp.:	180 Min.	0 Min.	0 Min.
Matrix solution:	10 Min.	10 Min.	0 Min.
Rendering:	16 Min.	16 Min.	16 Min.
TOTAL:	208 Min.	26 Min.	16 Min.

By changing only the viewing parameters, only the rendering portion of the computation was repeated to produce Image C, thus the savings in total time.

CONCLUSION

The extension of the radiosity solution to environments with occluded surfaces allows the simulation of complex scenes. All surfaces are treated as light sources and thus the global illumination effects are correctly modeled. The influence of occluded surfaces is contained within the solution for the form-factors, and thus shadows and shadow boundaries are properly reproduced.

It should be emphasized that the object space intensity computations are performed independent of the view position and direction. Thus, each phase of the procedure is independent of the modules which follow it. The independence of the algorithm steps permits rendering of dynamic sequences with little additional computation. The rendering can be repeated from different viewpoints without recomputing form-factors or radiosities. If the geometry of the environment remains static, the information to render changes in the lighting conditions is already contained in the form-factors. Thus lights can be turned on/off and object colors can be changed simply by resolving the matrices. The iterative solution converges rapidly, typically in six to eight interations, regardless of the size of the matrix. The entire process need be repeated only with changes in geometry.

The extension of the radiosity method to occluded environments permits the rendering of complex scenes, and thus should have great relevance to computer generated imagery. The computational expense is minimal compared to ray-tracing algorithms especially when rendering dynamic sequences within a static geometry.

A number of issues must be addressed in order to make the radiosity method both general and efficient. Since the accuracy/fineness of the numerical integration of the form-factor depends partially upon the area projected onto the hemi-cube, problems can occur due to aliasing for small projections. In general, objects whose projection onto the hemi-cube is small have little effect. However, for very bright patches such as light sources, the effect of the inaccuracy can have a substantial deleterious effect.

Generalizations must be made to accomodate curved surfaces. Polygonal approximation that would allow use of the same algorithms may suffice for the form-factor computations. However, the exact surface description should be used during rendering.

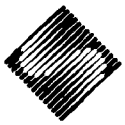
Since the size of the radiosity computation grows with the square of the number of patches, methods of approximation must be developed to keep the computation manageable as the complexity of the environment increases. Small variations in intensity across a patch have little effect on the overall global illumination. Rather than continue the subdivision of the patches to discern small radiosity variations, more detailed intensity information within a patch can be derived from a coarse patch radiosity solution. In this way the number of patches, and thus the computation time, can be kept to a minimum while allowing complex environments to be rendered. Texture mapping falls into the same category, and can be handled by treating a textured patch's effect on the environment as one average intensity during the radiosity solution.

Provisions must be made to incorporate specular and transparent surfaces into the rendering process or, better yet, into the initial radiosity formulation.

The radiosity method offers a fundamentally new approach to image synthesis by beginning from basic principles of conservation of energy. The radiosity method has been shown to be able to render complex environments. Radiosity should play a major role in future realistic image synthesis systems.

ACKNOWLEDGEMENTS

This research was conducted at the Cornell University Program of Computer Graphics and was supported by the National Science Foundation grant DCR8203979. As with all of the research efforts at CUPCG, this paper is the result of a team effort. Thanks must go to Cindy Goral and Professor Kenneth Torrance for their groundwork introducing radiosity to the computer graphics community. Thanks to David Immel, Philip Brock, and Channing Verbeck for help in software development, to Jutta Joesch, Lisa Maynes, and James Ferwerda for editing the text. Also to Gary Meyer and Holly Rushmeier for conduting experiments to verify the radiosity algorithms, to Janet Brown in preparing the text, and to Emil Ghinger for photography. Thank you, reviewers, for your helpful comments.



REFERENCES

[1] Amanatides, John. Ray Tracing with Cones. ACM Computer Graphics (Proceedings 1985), pp. 129-135.

[2] Cook, Robert. Distributed Ray Tracing. ACM Computer Graphics (Proceedings 1984), pp. 137-145.

[3] Foley, J. D. and Van Dam, A. Fundamentals of Computer Graphics. Addison-Wesley Publishing Co., 1982.

[4] Goral, Cindy M., Torrance, Kenneth E., Greenberg, Donald P., Battaile, Bennett, Modeling the Interaction of Light Between Diffuse Surfaces, ACM Computer Graphics (Proceedings 1984), pp. 213-222.

[5] Gouraud, Henri, Computer Display of Curved Surfaces, Ph.D. Dissertation, University of Utah, 1971.

[6] Hornbeck, Robert W., Numerical Methods. Quantum Publishers, 1975, pp. 101-106.

[7] Meyer, Gary W., Rushmeier, Holly E., Cohen, Michael F., Greenberg, Donald P., Torrance, Kenneth E., Assessing the Realism of Computer Graphics Images, submitted for publication, 1985.

[8] Newman, William M. and Sproull, Robert F., Principles of Interactive Computer Graphics, McGraw Hill, 1979.

[9] Phong, Bui Tuong, Illumination for Computer Generated Images, Ph.D. Dissertation, University of Utah, 1973.

[10] Siegel, Robert and Howell, John R., Thermal Radiation Heat Transfer, Hemisphere Publishing Corp., 1978.

[11] Sparrow, E. M. and Cess R. D., Radiation Heat Transfer, Hemisphere Publishing Corp., 1978.

[12] Weghorst, Hank, Hooper, Gary, and Greenberg, Donald P., Improved Computational Methods for Ray Tracing, ACM Transactions on Graphics, Jan, 1984, pp. 52-69.

[13] Whitted, Turner, An Improved Illumination Model for Shaded Display, Communications of the ACM, June, 1980.