VI.11

ACCURATE FORM-FACTOR

Filippo Tampieri
Cornell University
Ithaca, New York

The radiosity method for diffuse environments (Goral *et al*, 1984; Cohen and Greenberg, 1985) expresses the energy transfer between a source patch s and a receiving patch r as

$$\Delta B_r A_r = \rho_r B_s A_s F_{A_s \to A_r}, \qquad (1)$$

where

- ΔB_r is the unknown to be computed; it is the contribution of patch s to the radiosity of patch r, assumed constant across patch r;
- A_r is the area of patch r;
- ρ_r is the reflectivity of patch r;
- B_s is the radiosity of patch s, assumed constant;
- A_s is the area of patch s; and
- $F_{A_s \to A_r}$ is the form-factor from patch s to patch r and represents the fraction of energy leaving patch s and arriving at patch r.

Lying at the core of the radiosity formulation, the accurate computation of form-factors plays an important role in determining the quality of the radiosity solution.

This Gem presents a simple algorithm for the computation of accurate form-factors based on a number of ideas and techniques assembled from previously published work (Baum *et al.*, 1989; Wallace *et al.*, 1989; Hanrahan *et al.*, 1991). The algorithm computes the form-factor between

ISBN 0-12-409670-0 (IBM)
ISBN 0-12-409671-9 (Mac)

a finite-area polygonal source and a differential area on a receiving patch and can be used directly within the progressive refinement radiosity method (Cohen *et al.*, 1988).

If a differential area dA_r, centered about location x on patch r, is substituted for finite area A_r in Eq. (1), we obtain

$$\Delta B_r(x) dA_r(x) = \rho_r B_s A_s F_{A_s \rightarrow dA_r(x)} \tag{2}$$

Using the reciprocity relationship for form-factors, $A_s F_{A_s \rightarrow dA_r} = dA_r F_{dA_r \rightarrow A_s}$, and dividing both sides of Eq. (2) by dA_r, leads to

$$\Delta B_r(x) = \rho_r B_s F_{dA_r(x) \rightarrow A_s} = \rho_r B_s \int_{A_s} v(x, y) \frac{\cos \theta_r \cos \theta_s}{\pi r^2} dA_s(y), \tag{3}$$

where

$\Delta B_r(x)$ is the radiosity contribution of patch s to a differential area dA_r centered about point x;

$v(x, y)$ is a visibility term, 1 if point y is visible from x, 0 otherwise;

r is the distance between x and y;

θ_r, θ_s are the angles between the surface normals at x and y and the line connecting the two points; and

$dA_s(y)$ is the differential area centered about point y on patch s.

In the absence of occlusions, the visibility term is everywhere 1, and the area integral can be transformed into a contour integral using Stoke's theorem. This, in turn, can be evaluated analytically using the formula provided by Hottel and Sarofin (1967):

$$F_{dA_r(x) \rightarrow A_s} = \frac{1}{2\pi} \sum_{i=0}^{n-1} \frac{\cos^{-1}(R_i \cdot R_{i\oplus1})}{\|R_i \times R_{i\oplus1}\|} R_i \times R_{i\oplus1}) \cdot N_r, \tag{4}$$

where

$\oplus$ represents addition modulo n;

n is the number of vertices of source patch s;

R_i is the unit vector from point x on patch r to the ith vertex of patch s; and

N_r is the unit normal to patch r at point x.

Assuming the visibility term v were constant, it could be taken out from under the integral in Eq. (3), and the form-factor could be computed correctly using Eq. (4). Unfortunately, v is constant only when source patch s is either totally visible ($v(x, y) = 1$) or totally occluded ($v(x, y) = 0$) from the receiving differential area $dA_r(x)$. When partial occlusion occurs, however, approximating the visibility term with a constant function might result in a poor estimate of the form-factor.

A solution to this problem is not hard to find. By subdividing the source patch into smaller and smaller areas ΔA_j and using a different constant approximation to v for each ΔA_j, the visibility function can be approximated to any degree of accuracy, as can the integral in Eq. (3). Thus, the contribution of source s to receiving point x becomes

$$\Delta B_r(x) = \rho_r B_s \sum_{j=1}^{m} \left(V_j F_{dA_r(x) \to \Delta A_j}\right), \tag{5}$$

where V_j is the fraction of ΔA_j visible from x.

The following pseudo-code gives a recursive algorithm to compute ΔB_r:

```
    $\Delta B_r \leftarrow$ ComputeContribution($r(x), s, \epsilon_B, \epsilon_A$);
    ComputeContribution($r(x), s, \epsilon_B, \epsilon_l$)
        if CanCull($r(x), s$)
            then return 0;
            else begin
2.              $\epsilon_F \leftarrow \epsilon_B / (\rho_r B_s)$;
                return $\rho_r B_s$. ComputeFormFactor($r(x), s, \epsilon_F, \epsilon_A$);
                end;

    ComputeFormFactor($r(x), s, \epsilon_F, \epsilon_A$)
        begin
3.      $V \leftarrow$ ComputeVisibilityFactor($r(x), s$);
        if $V \leq 0$
```

```
4.                then return 0;
            else if V ≥ 1
                then begin
5.                  $F_{dA_r \to A_s} \leftarrow$ ComputeUnoccludedFormFactor($r(x), s$)
                    return $F_{dA_r \to A_s}$;
                    end;
            else begin
                $F_{dA_r \to A_s} \leftarrow$ ComputeUnoccludedFormFactor($r(x), s$);
6.              if $F_{dA_r \to A_s} \leq \epsilon_F$ or $A_s \leq \epsilon_A$
7                   then return $VF_{dA_r \to A_s}$;
                else begin
8.                  $(s_1, s_2, \ldots, s_m) \leftarrow$ SplitPatch($s$);
9.                  return $\sum_{j=1}^{m}$ComputeFormFactor($r(x), s_j, \epsilon_F, \epsilon_A$)
                    end;
                end;
            end;
```

Subroutine *ComputeContribution* takes as input the description of a differential area centered about point x on patch r, the description of a source patch s, a radiosity tolerance ϵ_B, and an area ϵ_A, and uses recursive subdivision of the source patch s to compute the radiosity contribution within an accuracy controlled by ϵ_B. The area term ϵ_A is used to prevent excessive subdivision such as could occur for a receiving point very close to an edge of the source.

The subroutine starts by calling subroutine *CanCull* to check whether receiving point x is oriented away from patch s or is behind it (step 1), and returns a null contribution if either of the conditions hold. If point x cannot be culled out of consideration, a tolerance ϵ_F is derived (step 2) so that a form-factor computed within precision ϵ_F yields a radiosity contribution accurate within the required precision ϵ_B. Subroutine *ComputeFormFactor* relies on an external routine to get an estimate of the visibility factor V (step 3); *ComputeVisibilityFactor* should return 0 for total occlusion, 1 for total visibility, and the fraction of source s that is visible from point x in the case of partial occlusion. The cases when V is either 0 or 1 are simple: The radiosity contribution is respectively equal to 0 (step 4) and to $\rho_r B_s$ times the unoccluded analytical form-factor (step 5), computed by subroutine*ComputeUnoccludedFormFactor* using Eq. (4).

The case of partial occlusion deserves more care: If the unoccluded analytical form-factor falls within the ϵ_F tolerance (step 6), then the radiosity contribution cannot possibly be larger than the desired accuracy ϵ_B and is estimated by the product of the visibility factor V and the computed unoccluded form-factor (step 7). If, instead, the unoccluded form-factor is larger than ϵ_F, the source is subdivided (step 8) and its contribution computed as the sum of the individual contributions of its parts (step 9). Also note that the recursive subdivision of the source is stopped if the area of the source falls below a specified threshold ϵ_A. This is done to prevent infinite recursion for receiving points along the corners and edges of the source, as suggested by Hanrahan *et al.* (1991).

The accurate computation of the visibility term V may not be a simple task. Such a computation could be carried out by projecting every polygon onto the source from the point of view of the receiving point and then computing the ratio of the sum of the unoccluded areas of the source to its total area. Unfortunately, this approach is prohibitively expensive, and faster methods must be used.

As an alternative to computing the visibility term V exactly, a reasonable approximation can be computed much more efficiently by tracing one or more shadow rays between the source and receiver and averaging their 0/1 associated visibilities (Hanrahan *et al.*, 1991.) The accuracy and efficiency of this technique is improved even further by testing for total occlusion or total visibility first, and using shadow rays only if neither case can be recognized with certainty.

Shaft culling (Haines and Wallace, 1991) provides a means of computing a list of potential occluders for a given source–receiver pair. If the list is empty, the source is totally visible; if an object in the list cuts the shaft and completely separates the source from the receiver, then the source is totally occluded; otherwise, the source is treated as partially occluded, and shadow rays are used as described above. Note that shaft culling also speeds up this last process by reducing the number of possible occluders that need to be tested against shadow rays.

See also G3, F.8.