



## A PROGRESSIVE RADIOSITY ALGORITHM BASED ON PIECEWISE POLYNOMIAL INTENSITY DISTRIBUTION

HUJUN BAO, XIAOGANG JIN and QUNSHENG PENG<sup>\*</sup>

State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, China

*e-mail:* zucad@zjpta.pub.net.cn

**Abstract**—A progressive radiosity algorithm based on piecewise polynomial intensity distribution is presented in the paper. Unlike the conventional radiosity method, radiosity across each patch is assumed to vary in a polynomial distribution. A generalized radiosity equation for scenes with non-diffuse surfaces is then set up and solved by adopting the progressive refinement technique. To improve the efficiency of the algorithm, an adaptive Monte Carlo algorithm incorporation with localization technique is employed to evaluate the form-factors from source patches to nodes. Experimental results demonstrate the potential advantage of our method for rendering complex scenes. © 1997 Elsevier Science Ltd

### 1. INTRODUCTION

The generation of global illumination effects caused by light interreflection between surrounding surfaces is one of the main issues for realistic image synthesis. The radiosity method, introduced into computer graphics from the field of radiative heat transfer, provides a powerful tool for simulating the propagation of light within a closed environment. As its solution is view-independent, this method is very suitable for walkthrough of a diffuse environment.

Since Goral *et al.* introduced the radiosity method into computer graphics in 1984 [1], this method has undergone rapid developments during the past 10 years. The traditional radiosity method assumes that the surfaces in the environment are ideal diffuse and the radiosity across each local patch is constant. The light energy distribution across each surface is described as a piecewise constant and the whole surface is subdivided into a series of small patches, then the radiosity equation for ideal diffuse environment is set up. Cohen and Greenberg employ a numerical technique called hemi-cube to calculate the form-factors between patches [2]. To provide useful results at an early stage, Cohen *et al.* proposed a progressive refinement radiosity approach [3]. Later, the original radiosity method was generalized to handle specular and transparency surfaces [4–8].

Although the traditional radiosity method proves to be effective, recent research shows that the resultant light energy distribution strongly relies on the subdivision of the scenic surfaces. This problem will get more serious if the environment contains high curvature surfaces. There are two solutions to the problem. The first way is to subdivide these surfaces into smaller patches until the radiosity

across each patch is nearly constant. The second way is to approximate the radiosity across each patch with a high order polynomial function. It is easy to know that the first way cannot solve the approximation precision problem entirely, and it will bring serious storage burden. Now more and more attention is paid on the second method, research in this direction mainly include finite element method [8–11] and radiosity approximation using other kinds of basis functions such as wavelet basis [12, 13]. The methods based on linear approximation proposed by Bao and Max are two of the earliest approaches [8, 9], they incorporate the interpolation function into the radiosity equation, and this makes it possible to obtain a more accurate radiosity solution with less patches. Moreover, Bao's method is generalized to a non-diffuse environment.

In this paper, we adopt piecewise polynomial to express the radiosity distribution across each patch, and the corresponding radiosity equation is set up accordingly. This equation is generalized to handle non-diffuse environment which contains ideal specular surfaces and transparency surfaces. As 2-degree approximation has the advantages of good approximation precision and less computation cost, a progressive refinement radiosity method based on 2-degree polynomial interpolation is presented as an example. In order to improve the computation efficiency, an adaptive localization technique is adopted to accelerate the calculation of form-factors.

### 2. RADIOSITY EQUATION BASED ON PIECEWISE POLYNOMIAL INTENSITY DISTRIBUTION

In a closed environment, the radiant light energy of differential surface area  $dS(x)$  containing sample point  $x$  is made up of its emitted energy and the reflected energy. That is,

$$B(x)dA(x) = E(x)dA(x) + \rho(x)H(x) \quad (1)$$

<sup>\*</sup> Author for correspondence.

where  $B(x)$  is the radiosity of differential surface area  $dS(x)$ ,  $dA(x)$  is the area of  $dS(x)$ ,  $\rho(x)$  is the diffuse reflectivity at point  $x$ ,  $H(x)$  is the total incident light energy to  $dS(x)$ , and  $E(x)$  is the self-emission radiosity. If the surface where  $x$  lies is a light source, then  $E(x) > 0$ , otherwise  $E(x) = 0$ .

By definition we know that  $H(x)$  is a function of radiosity  $B(x')$  ( $x' \neq x$ ) of the surrounding surfaces. Generally, only a part of the radiant energy leaving differential surface area  $dS(x')$  can arrive at  $x$ , we use form-factor  $G(x, x')$  to represent the fraction of energy leaving differential surface area  $dS(x')$  that arrives at differential surface area  $dS(x)$ . Then the contribution of  $dS(x')$  to  $dS(x)$  containing  $x$  is  $B(x')G(x, x')dA(x')$ , and  $H(x)$  can be expressed as

$$H(x) = \int_s B(x')G(x, x')dA(x') \quad (2)$$

By the definition of form-factor we know that  $G(x, x')$  can be expressed as

$$G(x, x') = V(dS(x), dS(x')) \frac{\cos \theta_x \theta_{x'}}{\pi r^2(x, x')} dA(x) \quad (3)$$

The relevant parameters are described in Fig. 1.  $V(dS(x), dS(x'))$  is the occlusion function.

Combining Equations (1), (2) and (3), we obtain the radiosity equation for general diffuse environment,

$$B(x) = E(x) + \rho(x) \frac{1}{dA(x)} \int_s B(x')G(x, x')dA(x') \quad (4)$$

The above system equation describes the energy distribution of the scenic surfaces for an ideal closed diffuse environment at its equilibrium status. If the solution to the equation can be calculated, the environment can be rendered by any hidden surface removal algorithms.

As the radiosity distribution in a scenic surface is a complex function, it is difficult to obtain its analytic expression. In order to solve this problem, the

traditional radiosity method divides scenic surfaces into a series of patches and assumes the radiosity across each patch is constant, thus a simple radiosity system equation is established. In this paper, we assume the environment is divided into  $N$  patches, the self-emission and the diffuse reflectivity of each patch are constant, but the radiosity across each patch is variant. We adopt piecewise polynomial to approximate the inner distribution of the radiosity function across each patch, then Equation (4) becomes

$$B_i(x) = E_i + \rho_i \frac{1}{dA_i(x)} \sum_{j=1}^N \int_{x' \in S_j} B_j(x')G(x, x')dA_j(x'), \quad i = 1, 2, \dots, N \quad (5)$$

where  $B_i(x)$  is the radiosity of the  $i$ th patch  $S_i$  containing point  $x$ . The other parameters are defined similarly, the parameters with sub index  $i$  are related to patch  $S_i$ . If the radiosity function  $B_j(x')$  of patch  $S_j$  is approximated by a polynomial function  $\tilde{B}_j(x')$ , we have

$$B_j(x') \approx \tilde{B}_j(x') = \sum_{l=0}^{M-1} B(j_l)N_l(x') \quad (6)$$

where  $N_l(x')$  is the node basis function in polynomial function space,  $M$  is the number of nodes and it is relevant to the degree of polynomial function  $B_j(x')$ . For example, if  $N_l(x')$  is a two dimensional linear function,  $M=4$ ; if  $N_l(x')$  is a two dimensional quadric function,  $M=8$ .

As the radiosity function of patch  $S_j$  is determined by  $B(j_l)$  ( $l=0, 1, 2, \dots, M-1$ ) completely, we can convert the unknown variables of Equation (5) into the radiosity functions at eight nodes of patch  $S_j$ . Substituting Equation (6) into Equation (5) we obtain

$$B(i_k) = E_i + \rho_i \sum_{j=1}^N \sum_{l=0}^{M-1} B(j_l)F_{S_i \rightarrow i_k}^l, \quad k = 0, 1, \dots, M-1 \quad (7)$$

where

$$F_{S_i \rightarrow i_k}^l = \frac{1}{dA(i_k)} \int_{x' \in S_i} N_l(x')G(x, x')dA_j(x') \quad (8)$$

Obviously it is a set of system equations with  $n$  unknown variables and  $n$  equations ( $n$  is the number of nodes in the scene), it can be solved by ordinary iterative approach.

Equation (7) is only suited to ideal diffuse environment. For general non-diffuse environment, specular patches and transparency patches as the transfer media of diffuse radiosity should be incorporated into the radiosity equation, then Equation (7) becomes

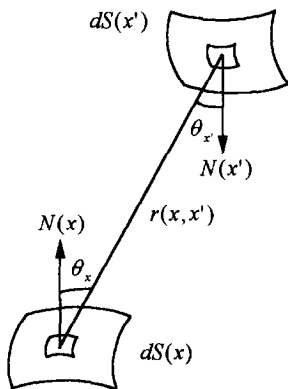


Fig. 1. Geometry of form-factor.

$$\begin{aligned}
B(i_k) = & E_i + \rho_i \sum_{j=1}^N \sum_{l=0}^{M-1} B(j_l) F_{S_j-i_k} \\
& + \frac{\rho_i}{dA(i_k)} \sum_{j=1}^N \int_{S_j} \left( b_j^{(r)(p)}(x') + b_j^{(t)(q)}(x') \right) G(x', i_k) dA(x') \\
& i = 1, 2, \dots, N; k = 0, 1, \dots, M-1
\end{aligned} \quad (9)$$

where  $b_j^{(r)(p)}(x')$  and  $b_j^{(t)(q)}(x')$  represent the indirect radiosity contributions to node  $i_k$  of patch  $S_j$  from the differential patch  $dS_p$  and patch  $dS_q$ , respectively, after they are specularly reflected and regularly transmitted by patch  $dS(x')$ . As these two variables are determined by other specular and transparency surfaces in the environment, they are defined recursively. Their expressions are

$$\begin{aligned}
b_j^{(r)(p)}(x') = & \rho_j^{(r)} V(dS_p, dS(x')) \\
& \left[ B_p(x'') + b_p^{(r)(p1)}(x'') + b_p^{(t)(q1)}(x'') \right] \quad (10)
\end{aligned}$$

$$N_l(x') = \begin{cases} 1, & \text{when } x' \text{ is node } l \\ 0, & \text{when } x' \text{ is other node} \\ \text{interpolating function of degree 2,} & \text{others} \end{cases}$$

$$\begin{aligned}
b_j^{(t)(q)}(x') = & \rho_j^{(t)} V(dS_q, dS(x')) \left( \frac{\eta_j}{\eta_q} \right)^2 \\
& \left[ B_q(x'') + b_q^{(r)(p2)}(x'') + b_q^{(t)(q2)}(x'') \right] \quad (11)
\end{aligned}$$

where  $\rho_j^{(r)}$  and  $\rho_j^{(t)}$  represent the specular reflectivity and the regularly transparency reflectivity of patch  $S_j$  respectively, the meanings of  $p_1, q_1, p_2, q_2$  are similar to that of  $p, q$ . Detailed deduction can be found in reference [8]. By now, we have established the radiosity equation for general environment based on piecewise polynomial intensity distribution.

### 3. SOLUTION TO THE RADIOSITY EQUATION

The solution to system Equation (9) can be obtained by the ordinary Gauss–Siedel iterative approach, and the form-factors can be calculated by the traditional hemi-cube method [2] or raytracing algorithm [14]. But as the time complexity of the above equation is  $O(n^2)$  and  $n \gg N$  ( $n$  is the number of nodes,  $N$  is the number of patches), solving Equation (9) using the Gauss–Siedel approach requires a large amount of main memory. In order to alleviate the memory requirement, we adopt a refined progressive refinement radiosity approach.

The progressive refinement radiosity algorithm provides the user with a useful approximative solution as early as possible and efficiently reduces the cost of storage. At each iterative step, a patch with greatest illumination effect to the environment is selected as the source patch which radiates energy to

the surrounding patches. This method reduces the storage requirement from  $O(N^2)$  to  $O(N)$ . Moreover, the environment of current status can be displayed after each iteration is finished, and this brings convenience in interactive design.

Since polynomial of degree 2 has good approximation precision and less computation cost, in the following discussion we restrict  $N_l(x')$  to a polynomial of degree 2 (higher degree cases can be dealt with similarly). The simplest choice of polynomial basis function  $N_l(x')$  of degree 2 is: 1,  $u$ ,  $v$ ,  $uv$ ,  $u^2v$ ,  $u^2$ ,  $u^2$ ,  $v^2$ , but this choice will bring trouble in subsequent processes. In this paper, we adopt the following method to construct basis function  $N_l(x')$ .

Without loss of generality we assume all the patches in the environment are quadrilaterals (triangles can be handled similarly), and the four edges of each quadrilateral are iso-parameter lines (see Fig. 2). We choose four vertices and the midpoints of each edge as the eight nodes, each node corresponds to a basis function, and  $N_l(x')$  satisfies

It is easy to prove that the constructed  $N_l(x')$  ( $l=0, 1, 2, \dots, M-1$ ) form a basis space. From the Lagrange interpolating theorem, the polynomial of degree 2 satisfying the above conditions can be expressed as

$$\begin{cases} N_0(x') = \frac{1}{4}(1-\zeta)(1-\eta)(-\zeta-\eta-1) \\ N_2(x') = \frac{1}{4}(1+\zeta)(1-\eta)(\zeta-\eta-1) \\ N_4(x') = \frac{1}{4}(1+\zeta)(1+\eta)(\zeta+\eta-1) \\ N_6(x') = \frac{1}{4}(1-\zeta)(1+\eta)(-\zeta+\eta-1) \\ N_1(x') = \frac{1}{2}(1-\eta)(1-\zeta^2) \\ N_3(x') = \frac{1}{2}(1+\zeta)(1-\eta^2) \\ N_5(x') = \frac{1}{2}(1+\eta)(1-\zeta^2) \\ N_7(x') = \frac{1}{2}(1-\zeta)(1-\eta^2) \end{cases} \quad (12)$$

where

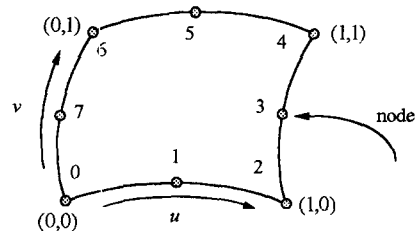


Fig. 2. The parametric definition of a patch.

$$\zeta = \frac{2(u - u_c)}{u_{j+1} - u_j}, \eta = \frac{2(v - v_c)}{v_{j+1} - v_j}, u_c = \frac{u_{j+1} + u_j}{2},$$

$$v_c = \frac{v_{j+1} + v_j}{2}, x' = x'(u, v)$$

The basic idea to solve Equation (9) using progressive refinement radiosity method can be described as follows: at each iteration step, we select the patch  $S_j$  with greatest unshot energy, the algorithm then calculates the direct energy contribution from patch  $S_j$  to every node in the scene, thus the diffuse incremental radiosity (the second term of Equation (9)) is determined. Then we calculate the non-diffuse incremental radiosity (the third term of Equation (9)) transmitted to this node through all specular patches and transparency patches caused by the incremental radiosity of other patches. This finishes a whole iteration. In the following we will discuss the energy transmission process in detail.

From Equation (7) we know the incremental radiosity of each node in the scene due to the radiation of patch  $S_j$  is

$$\Delta Rad(i_k) = \rho_i \sum_{l=0}^7 B(j_l) F_{S_j-i_k}^l \quad (13)$$

where  $i_k$  is the  $k$ th node in patch  $S_j$ . Obviously, the key to the problem is how to calculate the form-factor  $F_{S_j-i_k}^l$  efficiently.

It is worth pointing out that the above form-factor no longer possesses commutation, therefore it is impossible to calculate all the form-factor by establishing only a hemi-cube at the radiating source  $S_j$ . Thus we adopt the Monte Carlo method to calculate  $F_{S_j-i_k}^l$ ,

$$F_{S_j-i_k}^l = \frac{1}{dA(i_k)} \frac{1}{m} \sum_{h=1}^m \frac{N_l(x_h') G(i_k, x_h')}{p(x_h')}$$

where  $x_h'$  is the sample point in  $S_j$ ,  $m$  is the number of sample points,  $p(x')$  is the random sample density function of patch  $S_j$ .  $p(x')$  may be defined in various ways, here we simply choose  $p(x') = 1/A_j$ . Sample point  $x_h'$  can be located by the Jitter method and Poisson disk method. In the Jitter method,  $S_j$  is firstly uniformly subdivided according to the number of sample points,  $m$ , then a random point  $x_h'$  is generated at each differential patch  $e_h$ . The number of sample points  $m$  is determined by the following formula

$$m = \max \left( \left\lceil \frac{d\omega_j}{T_w} \right\rceil, \left\lceil \frac{B_{\max}}{T_B} \right\rceil \right) \quad (14)$$

where  $T_w$ ,  $T_B$  are prescribed solid angle and radiosity thresholds,  $B_{\max}$  is the greatest radiosity contribution from patch  $S_j$  to node  $i_k$ , and it can be estimated according to Equation (14). Therefore

$$F_{S_j-i_k}^l = \frac{A_j}{dA(i_k)} \frac{1}{m} \sum_{h=1}^m N_l(x_h') G(i_k, x_h') \quad (15)$$

Note that the occlusion test function  $V(x_h', i_k)$  in  $G(i_h', i_k)$  can be computed by intersecting the line connecting  $x_h'$  and  $i_k$  with the scenic surfaces (not patches) directly. All the acceleration method for ray/scene intersection can be adopted in this process.

The key to calculate the non-diffuse term lies in the calculation of  $b_j^{(r)(p)}$  and  $b_j^{(t)(q)}$ . As  $b_j^{(r)(p)}$  and  $b_j^{(t)(q)}$  are defined recursively, we use raytracing to calculate them. From Equations (10) and (11) we know that they can be calculated by the following steps: (1) from the current node a ray is shot to the random sample point  $x_h'$  of each micropatch  $e_{s,l}$  of the shooting patch  $S_j$ ; (2) for each ray, the nearest points  $x_p$  and  $x_q$  (lie in patches  $p$  and  $q$ , respectively) are calculated along the direction of reflected ray and refracted ray; (3) the diffuse incremental radiosities at points  $x_p$  and  $x_q$  contributed by patches  $p$  and  $q$  are calculated using Equation (6), and the radiosity contributions to node  $i_k$  from shooting patch are obtained by multiplying them with the corresponding specular reflectivity and regular transparency reflectivity; (4) the above steps continue until a ray hits an ideal diffuse surface or the ray trace level reaches the predefined maximum level.

Note that in the above progressive radiosity procedure the patch with greatest unshot energy is always selected as the shooting source patch at each step. As the iteration continues, the unshot energy of the shooting patch is getting smaller and smaller, and the affected region of the shooting patch is getting less and less and finally tends to be none. This demonstrates that the shooting patch contributes only to a very small local region after some iterations, the energy contribution to the nodes outside of the local region can be neglected. As the local affected region can be determined with little cost, the efficiency of the above progressive radiosity can be greatly improved.

For simplicity, we approximate the local affected region of the shooting patch with a sphere centered at the center of the patch. Obviously, the point on the sphere receiving the greatest radiosity contribution from the shooting patch lies on the line along the normal of the patch at its center, the energy contribution is

$$B_{\max} = FB_{\text{source}}$$

where  $B_{\text{source}}$  is the average unshot radiosity of the shooting patch,  $F$  is the form-factor from the shooting patch to this point and can be approximated by (see Fig. 3)

$$F \approx \frac{A_{\text{source}}}{\pi r^2 + A_{\text{source}}}$$

where  $r$  is the radius of the affected region of current

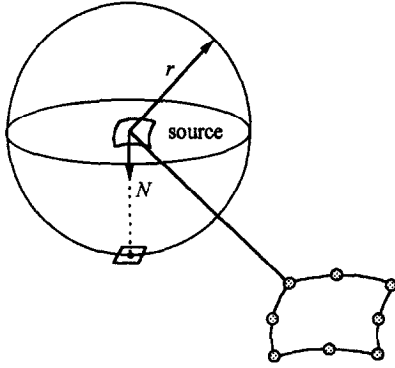


Fig. 3. The determination of the affected region of the shooting patch.

shooting patch, it can be determined by the following formula

$$r = \sqrt{\frac{(B_{source} - \epsilon)A_{source}}{\pi\epsilon}}$$

where  $\epsilon$  is the prescribed threshold.

From above discussion we know that if the distance from the center of the shooting patch to a node is greater than  $r$ , the radiosity contribution to the node from the shooting patch can be neglected. As  $B_{source}$  is getting smaller and smaller, so is  $r$ , thus the above progressive refinement radiosity becomes more and more efficient. Because a shooting patch will shoot no energy to its back face, back-culling can be employed to improve the efficiency.

The data structure of a node is

```
struct node {
    float position [3], Normal [3]; /*the position and normal of a node*/
    float B; /* the radiosity of current node */
    float ΔB; /* accumulated unshot energy */
    float ΔRad; /* incremental radiosity after each iteration */
    unsigned done; /* a flag indicates that a node has been computed */
}
```

By now we can depict the progressive refinement radiosity algorithm for solving Equation (9).

```
/*initialization*/
ρ = Σi=1N ρi Ai / Σi=1N N Ai
Ambient = ρ Σi=1N Ei Ai / Σi=1N Ai
for (each node ik in the environment) {
    set NodePtr(ik) as the pointer which points to node ik;
    NodePtr(ik) → B = Ei + ρi Ambient;
    NodePtr(ik) → ΔB = Ei;
}
Until convergence {
    select patch Sj with greatest Aj Σi=07 NodePtr(ik) → ΔB as the shooting patch;
    calculate the square of the radius of its local affected region r2;
    for (each node ik in the environment) {
        NodePtr(ik) → ΔRad = 0;
        if (the square of the distance between ik and center of the shooting patch is less than r2 && ik lies in the positive half space of the shooting patch) {
```

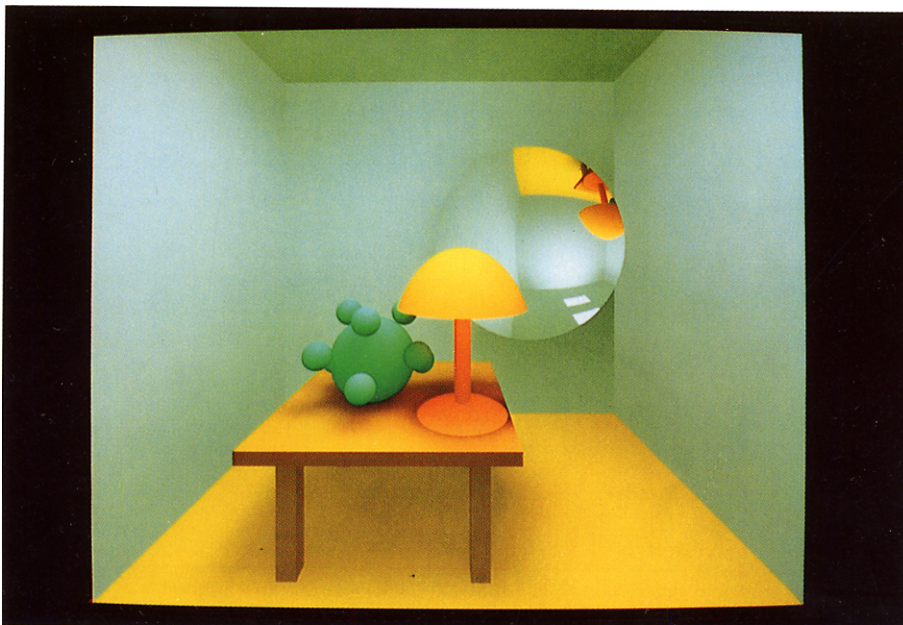


Fig. 4. A test environment of curved surface with diffuse energy distribution only.

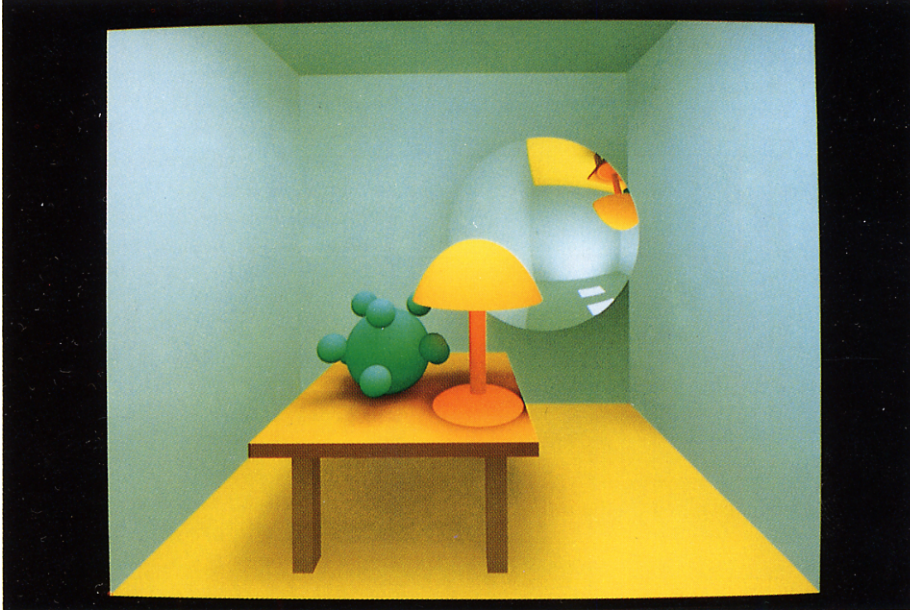


Fig. 5. The same environment as Fig. 4 with indirect light energy transfer caused by specular reflections.

```

NodePtr(ik)→done=FALSE;
else
  NodePtr(ik)→done=TRUE;
}
for (each patch Sj in the environment){
  for (each node ik of patch Sj){
    if(NodePtr(ik)→done=FALSE{
      determine the number of sample points m
      in patch Sj according to equation (14),
      calculate each random sample point x'h
      and normal to the point, and compute all
      FSj-ikl (l=0, 1, 2, . . . , 7);
      NodePtr(ik)→done=TRUE;
      NodePtr(ik)→ΔRad=ρi∑l=07FSj-iklNode-
      Ptr(ik)→ΔB
    }
  }
}
for (each specular or transparency patch
Sj){
  calculate sample points x's,t on patch Sj
  and the normals at these points, calculate
  the area of each micropatch es,t;
  set the done flag in node data structure be
  FALSE;
  SRad=0.0;
  for (each node ik in the environment) {
    if(NodePtr(ik)→done=FALSE)
      for (each micropatch es,t on patch Sj){
        compute G(ik, x's,t);
        start from ik, shoot a ray to point
        x's,t and trace into the environ-
        ment, determine bj(r)(p) and
        bj(t)(q);
        SRad=SRad+G(ik, x's,t)
        (bj(r)(p)+bj(t)(q));
      }
      NodePtr(ik)→done=TRUE;
      NodePtr(ik)→ΔRad+=ρiSRad;
      NodePtr(ik)→B+=ρiSRad;
    }
  }
}
for (each node ik in the environment){
  NodePtr(ik)→ΔB+=NodePtr(ik)→ΔRad;
  NodePtr(ik)→B+=NodePtr(ik)→ΔRad+ρi-
  ΔAmbient;
}
for (each patch Sj in the environment){
  ΔG+=(∑k=07NodePtr(ik)→ΔRad/8;

```

Table 1. CPU time for generating Figs 4–6

Iteration: 200	Environment subdivision	Radiosity calculation	Rendering	Total time
Fig. 4	9"	49'40"	9'50"	59'39"
Fig. 5	9"	62'20"	9'50"	72'19"
Fig. 6	33"	262'41"	24'15"	287'29"



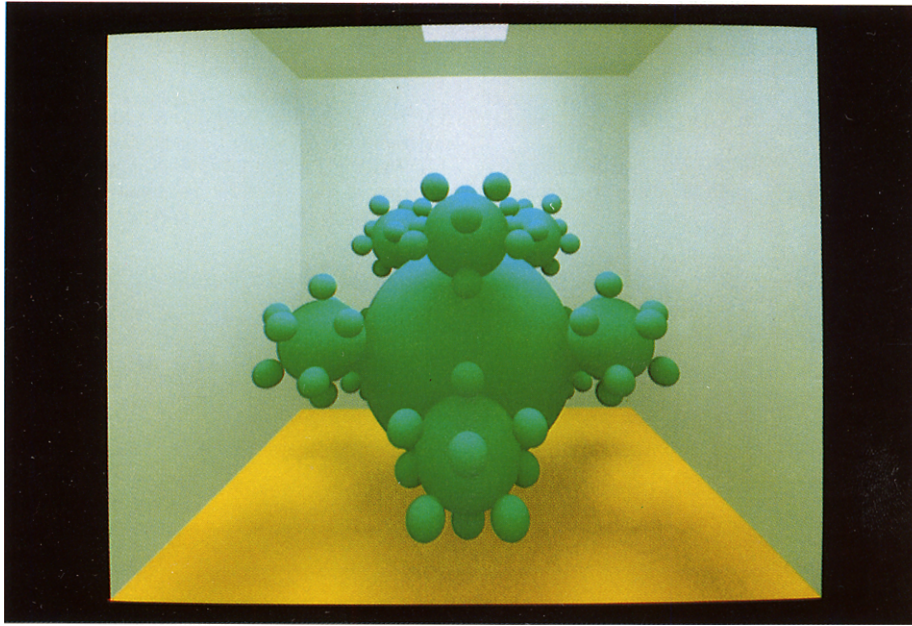


Fig. 6. Another test environment of curved surface with diffuse energy distribution only.

```

 $\Delta \text{Ambient} = \text{RAG} \sum_{i=1}^N A_i;$ 
set the unshot radiosity of each node of
shooting patch  $S_j$  to 0;
}

```

#### 4. IMPLEMENTATION

The above algorithm has been implemented on an SGI 4D-35 workstation at the State Key Lab of CAD & CG, Zhejiang University. Figures 4 and 5 adopt the same test environment. Figure 4 concerns only the diffuse energy distribution, in Fig. 5 indirect energy contribution via mirrors are taken into account. From the two pictures we can see the difference of energy distribution on the floor. Figure 6 is another test scene rendered with our algorithm, the environment consists of 91 spheres, 9 planes and 3 light sources. 200 iterations are used for progressive refinements. Ordinary ray tracing algorithm is adopted in the rendering phase and  $2 \times 2$  super-sampling is employed to eliminate aliasing. The thresholds of solid angle and maximum energy is predefined as  $0.001256(T_w)$  and  $2.5(T_B)$ , respectively. The CPU time for generating the three images are listed in Table I. The resolution of the three images is  $1280 \times 1024$ .

#### 5. CONCLUSION

We have presented a new progressive radiosity algorithm which assumes that the radiosity across each patch varies in a polynomial distribution. A general radiosity equation for scenes with non-diffuse surfaces is then set up and solved by adopting a progressive refinement technique. Adaptive Monte

Carlo random sampling technique is employed to calculate the form-factor from a patch to a node, and we have also developed the technique of light energy transfer localization, which successfully reduces the computational cost of solving the radiosity equation. With our method, the surfaces of the environment can be subdivided in a coarser manner than with the traditional radiosity method while keeping the same precision and hence suitable for complex environment rendering.

#### REFERENCES

1. Goral, C. M., Torrance, K. E., Greenberg, D. P. and Battaile, B., Modeling the interaction of light between diffuse surfaces. *Computer Graphics*, 1984, **18**, 213–222.
2. Cohen, M. F. and Greenberg, D. P., The hemi-cube: a radiosity solution for complex environments. *Computer Graphics*, 1985, **19**, 31–40.
3. Cohen, M. F., Chen, S. E., Wallace, J. R. and Greenberg, D. P., A progressive refinement approach to fast radiosity image generation. *Computer Graphics*, 1988, **22**, 75–84.
4. Immel, D. S., Cohen, M. F. and Greenberg, D. P., A radiosity method for non-diffuse environment. *Computer Graphics*, 1986, **20**, 133–142.
5. Wallace, J. R., Cohen, M. F. and Greenberg, D. P., A two-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods. *Computer Graphics*, 1987, **21**, 311–320.
6. Sillion, F. and Puech, C., A general two-pass method integrating specular and diffuse reflections. *Computer Graphics*, 1989, **23**, 335–344.
7. Shao, M. Z., Peng, Q. S. and Liang, Y. D., A new radiosity approach by procedural refinements for realistic image synthesis. *Computer Graphics*, 1988, **22**, 93–101.

8. Bao, H. J. and Peng, Q. S., A progressive radiosity algorithm for scenes containing curved surfaces. *Computer Graphics Forum*, 1993, **12**, 81–90.
9. Max, N. L. and Allison, M. J., Linear radiosity approximation using vertex-to-vertex form factors. In *Graphics Gems III*, ed. D. Kirk. Academic Press, San Diego, 1992.
10. Zatz, H. R., Galerkin radiosity: a higher order solution method for global illumination. *Computer Graphics*, 1993, **27**, 213–216.
11. Troutman, R. and Max, N. L., Radiosity algorithm using higher order finite element methods. *Computer Graphics*, 1993, **27**, 209–212.
12. Gortler, S. J., Schroder, P. and Cohen, M. F., Wavelet radiosity. *Computer Graphics*, 1993, **27**, 221–230.
13. Yu, Y. Z. and Peng, Q. S., Multiresolution B-spline radiosity. *Computer Graphics Forum*, 1995, **14**, 285–298.
14. Wallace, R., Elmquist, K. A. and Haines, K. A., A ray tracing algorithm for progressive radiosity. *Computer Graphics*, 1989, **23**, 315–324.