

An Efficient Radiosity Solution for Bump Texture Generation

Hong Chen and En-Hua Wu

Institute of Software, Academia Sinica
P. O. Box 8718, Beijing 100080, China

Abstract

The development of global illumination and texture generation makes it possible to produce the most realistic images. However, it is still difficult or deficient so far to simulate bump texture effects while the interreflection of light being modeled by the present ray tracing or radiosity methods. A method of bump texture generation, being incorporated into the process of radiosity solution, is presented in the paper. The method is characterized by introduction of a *perturbed radiosity map*, established in the context of either progressive radiosity or standard radiosity solution. To calculate the perturbed radiosity, a concept of *perturbed form-factors* is proposed, and the algorithms for evaluating the perturbed form-factors are also given. As a result, a bilinear-interpolation shading scheme for perturbed surfaces is provided, and the texturing method is easily added to a newly improved solution of progressive refinement radiosity for non-diffuse environment.

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

General Terms: Algorithms.

Additional Key Words and Phrases: perturbed radiosity map, perturbed form-factor, perturbed hemi-cube, bumpy texture, radiosity, interreflection, progressive refinement.

1. Introduction

The most realistic and attractive computer generated images are those that contain a large amount of visual complexity and detail. Ray tracing and radiosity methods are able to produce highly realistic images by faithful simulation of light energy exchange between surfaces. However, if an image is rendered only by taking illumination model into

account, it will look artificial due to the extreme smoothness of surfaces. To overcome this shortage, the surface complexity should be increased. Comparatively, texturing is an acceptable method that can increase the surface detail by a relatively low cost. Generally speaking, the surface texture mapping approaches come into two main categories: color mapping and bump mapping.

The conventional ray tracing method simulates light reflection and refraction for specular and transparent surfaces, and produces direct illumination and shadow effects[23]. The diffuse interreflection in the ray tracing is simply treated as a constant "ambient" term, which makes images obviously computer-generated, as the interreflection has significant effect on the lighting of a scene.

There have been many approaches[2][3][4][5][6][14] which incorporate the texture generation process directly into reflection calculation. These approaches are suitable for rendering pictures via standard ray tracing method. But within the standard ray tracing methodology, the texture solution for diffuse interreflection is by no means viable due to the inherent nature of standard ray tracing.

In 1988, Ward et al. introduced a new ray tracing method of modeling (diffuse) interreflection[22]. By their method, indirect illuminance is averaged over surfaces from a small set S of computed values to avoid calculation at each pixel. The members of the set S are determined by the split sphere model which estimates the illuminance gradient on each surface using the illuminance coherence. But such split sphere model only relates the illuminance gradient to the scene geometry without considering the lighting distribution. To maintain accuracy, when the orientation of a surface, such as a wrinkled surface, changes rapidly, the computation of indirect illuminance increases at a very high rate. Therefore, this method is not efficient for simulating textures, in particular the bumpy texture[3], with diffuse interreflection effect.

The radiosity method, based on the principle of heat transfer[19], is well suited to calculating diffuse interreflection[12]. In 1985 and 1986, Cohen, Greenberg et al. developed the standard radiosity method, in a diffuse environment[8][9]. Various radiosity methods were developed in the following years for rendering pictures in a non-diffuse

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

environment[20][16][17][18]. The progressive refinement radiosity method proposed by Cohen et al. in 1988[10] has made it possible for radiosity to be employed interactively for image synthesis in a complex environment.

A diffuse interreflectional texture algorithm based on a reformed colour texture mapping technique was realized in Cohen's standard radiosity approach[9]. However, no efficient technique available so far for simulating the bump mapping texture[3] in an interreflectional environment within the framework of radiosity or ray tracing methods.

In the following part of the paper, we will present a radiosity solution for bump mapping texture rendering, based on illuminance coherence. The technique has been realized without significant degradation of the radiosity methods utilized. Efficiency is obtained with an introduction of so called *perturbed radiosity map* and *perturbed form-factor*. Combined with an adapted progressive refinement radiosity method for non-diffuse environment[7], a general rendering method, which has a highly practical value, is established.

2. Bumpy Mapping Texture by Perturbed Radiosity Map

The slight change of a surface normal direction will result in variation in the intensity distribution of the reflected light. Based on this principle, Blinn first made a proposal for simulating wrinkled surfaces by perturbing surface normals. This is popularly known as "bumpy mapping"[3]. He developed a method for generating the direction and value of the normal perturbations from a bump function $F(u, v)$ of the surface parameters. $F(u, v)$ is stored as a two-directional lookup table indexed by u and v . By moving the parameterized surface $\vec{P}(u, v)$ a small value $F(u, v)$ in the direction of the surface normal \vec{N} , new normal \vec{N}' is calculated and the bumpy surface is produced in terms of \vec{N}' .

By the shading scheme of radiosity method, the radiosity at a point within a patch is calculated by bilinear interpolation from the radiosities at the patch vertices, and thus the true surface normal at the visible point has not been utilized directly during the rendering process. Therefore, the traditional bump mapping technique is unable to be added into radiosity solution.

We notice that within a relatively small range, the surfaces which have the same normal will have their illuminance more or less the same. Based on this phenomenon of *illuminance coherence*, we are able to design a shading scheme to calculate the perturbed radiosity on a perturbed patch. Briefly speaking, if the radiosity $B(\vec{N}', \vec{P}_v)$ at a patch vertex P_v can be calculated and recorded in a *perturbed radiosity map* for different perturbed directions of \vec{N}' , then The perturbed radiosity $B(\vec{N}', \vec{P})$ at any point \vec{P} on the patch can be derived from the traditional method of bilinear interpolation of the radiosities $B(\vec{N}', \vec{P}_v)$ at vertices.

If the patch is not perturbed, \vec{N}' is identical to \vec{N} , $B(\vec{N}, \vec{P})$ can be determined just as in the standard radiosity method.

Next, in Section 2.1, before going through the detailed discussion of texture generation, we first review an implementation of two-pass method for non-diffuse environments combining an adapted progressive radiosity and ray tracing methods. It was taken to be the bed for our texture generation. A description is given in Section 2.2 on how to evaluate the perturbed diffuse radiosity, which is followed in Section 2.3 by derivation of a method for the evaluation of perturbed form-factors.

2.1 Progressive Refinement Radiosity for Non-diffuse Environment

Within a diffuse environment, the radiosity of a patch, say i , is given by:

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j F_{ji} A_j / A_i \quad <1>$$

where B_i is radiosity of patch i ($watts/m^2$),
 E_i is emission of patch i ($watts/m^2$),
 ρ_i is reflectance of patch i ,
 A_i is area of patch i (m^2),
 F_{ji} is form-factor from patch j to patch i ,
 N is number of discrete patches in the environment.

The equation <1> suggests that the energy on patch i is the result of gathering those from the environment. In this standard radiosity method, energy distribution in the environment may be simply obtained by solving the set of simultaneous equations derived from equation <1>. However, establishment of the coefficient matrix requires calculating and saving form-factors between all patch pairs so that a computation and storage expense of $O(N^2)$ is required.

The method of progressive refinement radiosity has overcome the shortcoming above. The solution proceeds in a series of refinement steps by shooting radiosities from a shooting patch with greatest energy at each step. It is efficient since only one hemi-cube on the shooting patch need to be established at each iterative step, so that the time and space trade-off is only $O(N)$.

When the progressive refinement radiosity method[10] was proposed in 1988, it was constrained to diffuse environment. In 1989, an extended two-pass method for non-diffuse environment, capable of being adapted into a progressive process, was provided by Sillion and Puech[18]. In their method,

Do until convergence

Select patch with greatest unshot energy as shooting patch;

Compute form-factors from shooting patch to all other elements by hemi-cube algorithm;

If shooting patch is non-diffuse, compute delta irradiations;

Based on form-factors and delta irradiations, add contribution of the shooting patch to radiosity of all elements and irradiation of non-diffuse elements.

Instead of directly using the specular form-factors or extended form-factors like in methods in [16][17][18][21], in our method, when a selected shooting patch is non-diffuse, apart from calculating the standard form-factors by the newly constructed hemi-cube, delta irradiations are evaluated from the increment of irradiation which has been recorded in a queue from previous calculations. The delta irradiations are then utilized for calculating the specular portion of the shooting to other patches.

When the number of non-diffuse patches is relatively small compared to the number of diffuse patches, and the technique of subdivision[9][10] is employed, the storage cost of irradiation queues is low.

The method above is based on hemi-cube algorithm, so it is readily implemented in terms of hardware Z-buffer. In contrast, the method of ray tracing is now hardly implemented by hardware, particularly when a complex reflectance function and an advanced ray tracing algorithm such as distributed ray tracing are utilized. Therefore, our method, with the hemi-cube algorithm assisted by hardware on the workstation such as SGI IRIS 4D/20, is more efficient than the methods like that described in [18].

2.2 Perturbed Radiosity

As shown in Fig. 2, the plane $PL(i, \vec{N}')$ is through the center O_i of the patch i and its normal vector is the perturbed normal \vec{N}' . Because the value of the perturbation function $F(u,v)$ upon the patch i is small, we can assume that the solid angle $d\omega_{ji}$ is not affected by the perturbation. The Radiant flux Φ_{ji} is shooting out from patch j to patch i within the solid angle $d\omega_{ji}$. The area on the plane $PL(i, \vec{N}')$ which is subtended by the solid angle $d\omega_{ji}$ forms a new patch i' , referred to as perturbed patch i' , or *P-patch* i' . Note that P-patch i' and its area A_i' is affected by the perturbed normal vector \vec{N}' .

The form-factors between patch i and j have a reciprocity relationship[8]:

$$A_i F_{ij} = A_j F_{ji} \quad \langle 7 \rangle$$

where A_i is the area of patch i and A_j is the area of patch j ;

F_{ij} is the form-factor from patch i to patch j ;

F_{ji} is the form-factor from patch j to patch i .

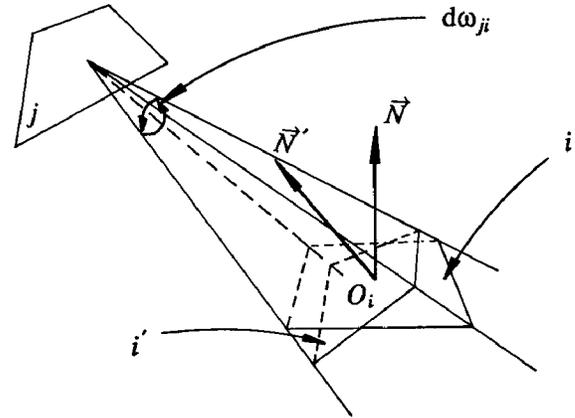


Fig. 2 Assumptive Perturbed Patch

Define the form-factor from P-patch i' to patch j , F'_{ij} , a function of \vec{N}' , as perturbed form-factor. Since the solid angle $d\omega_{ji}$ is constant, the form-factor from patch j to P-patch i' is equal to F_{ji} . So

$$A_i' F'_{ij} = A_j F_{ji} \quad \langle 8 \rangle$$

From $\langle 7 \rangle$ $\langle 8 \rangle$, we have

$$\frac{A_i'}{A_i} = \frac{F_{ij}}{F'_{ij}}$$

The irradiance $H'_{ji}(\vec{N}')$ of P-patch i' from patch j is:

$$H'_{ji} = \frac{\Phi_{ji}}{A_i'} = \frac{\Phi_{ji}}{A_i} \frac{F_{ij}}{F'_{ij}} = H_{ji} \frac{F_{ij}}{F'_{ij}} \quad \langle 9 \rangle$$

Similar to equation $\langle 2 \rangle$, the diffuse radiosity $B_i'(\vec{N}')$ of P-patch i' is

$$\begin{aligned} B_i' &= E_i + k_i \rho_i \pi \sum_{j=1}^N H'_{ji} \\ &= E_i + k_i \rho_i \pi \sum_{j=1}^N H_{ji} \frac{F_{ij}}{F'_{ij}} \end{aligned} \quad \langle 10 \rangle$$

It is clear from $\langle 10 \rangle$ that, to obtain the perturbed radiosity (B_i') of patch i , we have to know, for each patch j , the irradiation H_{ji} , and the form-factors F'_{ij} , F_{ij} . In other words, the perturbed radiosity on one patch is *gathered* from all other patches in the environment. Therefore, the task of perturbed radiosity evaluation has to be taken by a postprocess while the formula $\langle 10 \rangle$ is utilized. Latter on in the paper, we will derive another method to allow the evaluation of perturbed radiosity being performed at each iterative step in the process of progressive refinement solution with a *shooting* way.

Evaluation of H_{ij} has been described in Section 2.1. We now describe how to calculate the perturbed form-factors F'_{ij} .

2.3 Evaluation of Perturbed Form-Factor

Because the perturbation is small in comparison with the size of perturbed patch, it is reasonable to suppose that the index of visible patch preserved in pixels on imaginary hemi-cube $HM(\vec{N}_i)$ of patch i is not affected by perturbation. In another word, the hemi-cube of P-patch i' is the same as $HM(\vec{N}_i)$.

Note that the portion of $HM(\vec{N}_i)$ behind the P-patch i' should be discarded when calculating perturbed diffuse form-factor. So the perturbed hemi-cube $HM(\vec{N}'_i)$ is the fraction of $HM(\vec{N}_i)$ excluding the shadowed part (Fig. 3).

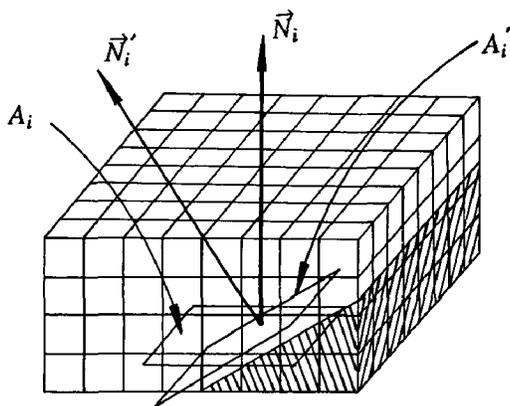


Fig. 3 Perturbed Hemi-cube

The form-factor between patches i and j is [8]:

$$F_{ij} = \frac{1}{A_i} \iint_{A_i, A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij} dA_j dA_i \quad <11>$$

By using the hemi-cube algorithm, F_{ij} is evaluated from the sum of delta form-factors:

$$F_{ij} = \sum_{p \in M} \Delta F_p$$

where ΔF_p is the delta form-factor associated with pixel P on hemi-cube $HM(\vec{N}_i)$;

M is the set of hemi-cube pixels covered by projection of patch j on hemi-cube $HM(\vec{N}_i)$;

$$\begin{aligned} \Delta F_p &= \frac{\cos\theta_p d\omega_p}{\pi} \\ &= \frac{\vec{N} \cdot \vec{R}_p}{|\vec{N}| |\vec{R}_p|} d\omega_p / \pi \end{aligned} \quad <12>$$

where θ_p is the angle between the normal vector \vec{N} and the direction vector \vec{R}_p from the center of the

patch i to the center of pixel P ;

$d\omega_p$ is the solid angle subtended by pixel p , and it is independent of normal vector \vec{N}_i .

Similarly, the perturbed form-factor can be evaluated by perturbed hemi-cube $HM(\vec{N}'_i)$:

$$F'_{ij} = \frac{1}{\pi} \sum_{p \in M'} \frac{\vec{N}' \cdot \vec{R}_p}{|\vec{N}'| |\vec{R}_p|} d\omega_p \quad <13>$$

where M' is the set of pixels covered by projection of patch j on perturbed hemi-cube $HM(\vec{N}'_i)$.

Delta perturbed form-factor is:

$$\Delta F'_p = \frac{\vec{N}' \cdot \vec{R}_p}{|\vec{N}'| |\vec{R}_p|} d\omega_p / \pi \quad <14>$$

Given \vec{N}' , to calculate F'_p for every pixel p on perturbed hemi-cube according to <14>, a dot product including three multiplications and two additions, plus another multiplication is required. The computation cost is therefore considerable. If the property of scan line coherence is utilized, the dot product will be reduced to an addition calculation. It still requires an $O(n)$ multiplication and an $O(n+p)$ addition, where n is the number of pixels on the hemi-cube $HM(\vec{N}_i)$ and p , greater than $0.5n$, is the number of pixels on the perturbed hemi-cube $HM(\vec{N}'_i)$. Because n is often very large, the latter method is still inefficient. In the following, we propose an efficient method based on an idea similar to coordinate division, to decrease a large amount of computation.

Note that

$$\vec{N}' \cdot \vec{R}_p = \sum_{k=x,y,z} [(\vec{R}_p \cdot \vec{I}_k)(\vec{I}_k \cdot \vec{N}')] \quad <15>$$

where $\vec{I}_x = [1 \ 0 \ 0]$, $\vec{I}_y = [0 \ 1 \ 0]$, $\vec{I}_z = [0 \ 0 \ 1]$.

Denote

$$w_k = \frac{\vec{I}_k \cdot \vec{N}'}{|\vec{N}'|}, \quad k=x,y,z \quad <16>$$

and define the delta form-factors in three directions as:

$$\Delta F_{pk} = \frac{\vec{R}_p \cdot \vec{I}_k}{|\vec{R}_p|} d\omega_p / \pi, \quad k=x,y,z \quad <17>$$

We obtain the delta perturbed form-factor

$$\Delta F'_p = w_x \Delta F_{px} + w_y \Delta F_{py} + w_z \Delta F_{pz} \quad <18>$$

which shows that in each item only w_k ($k=x,y,z$) but ΔF_{pk} ($k=x,y,z$) depends on the normal vector \vec{N}' . Thus, the perturbed form-factor

$$\begin{aligned} F'_{ij} &= \sum_{p \in M'} \Delta F'_p \\ &= w_x \sum_{p \in M'} \Delta F_{px} + w_y \sum_{p \in M'} \Delta F_{py} + w_z \sum_{p \in M'} \Delta F_{pz} \end{aligned}$$

$$= w_x F_{ijx}' + w_y F_{ijy}' + w_z F_{ijz}' \quad <19>$$

By traversing the $HM(\vec{N}_i)$, we obtain F_{ijx} , F_{ijy} and F_{ijz} , the global summations of the delta form-factors in three directions respectively. F_{ijx} , F_{ijy} and F_{ijz} are calculated only once because they are equivalent for all the possible perturbed normal vectors. For a given perturbed normal vector \vec{N}' , the corresponding w_x , w_y and w_z are calculated from the equation <16>. As shown in Fig. 3, the shadowed area is not bigger than the area of perturbed hemi-cube $HM(\vec{N}_i')$, therefore, to calculate F_{ijx}' , F_{ijy}' and F_{ijz}' , we only need calculate the three abandoned summations of delta form-factors by traversing the shadowed area, and subtract these abandoned summations from the corresponding global summations F_{ijx} , F_{ijy} and F_{ijz} respectively. Finally, the perturbed form-factor F_{ij} can be evaluated by <19>.

For non-perturbed patch, $\vec{N}' = \vec{N} = [0,0,1]$ and $w_x = w_y = 0$, $w_z = 1$, F_{ij} is only related to ΔF_{pz} : $F_{ij} = \sum_{p \in M} \Delta F_{pz}$.

Fig. 4a to 4f show a test example, produced by progressive radiosity with 60 iterations. The resolution of hemi-cube is 100 by 100. Fig. 4a took 92 seconds, without texture generation, and Fig. 4b. took 27 seconds of extra time for texture generation (independent of iteration), with nine extra hemi-cubes built over the perturbed elements on the wrinkled surface of the green box on the table, and 16 directions (mentioned latter) sampled for the evaluation of perturbed form-factors by means of perturbed hemi-cube algorithm.

Obviously, the method above has been derived from the view point of radiosity gathering, and can be readily applied in the context of various radiosities such as standard radiosity[9], two-pass radiosity[20] or Shao's method[16], where establishment of additional hemi-cube for perturbed patches is unnecessary, therefore, the cost of texturing operation is low.

However, if the mentioned technique is to be applied to the progressive radiosity method, additional form-factors should be particularly calculated. Therefore, when the texture generation is to be performed at each iterative step, the cost will become quite expensive unless the amount of perturbed patches is small. This fact prohibits the technique above from being effectively utilized in a complex environment with progressive radiosity approach. Of course, the technique could be used in a postprocess behind the termination of refinement process, but in which case the advantage of progressive radiosity would have not been utilized for texturing purpose. In that follows, we suggest an approximate bumpy texture method based on the idea of radiosity shooting.

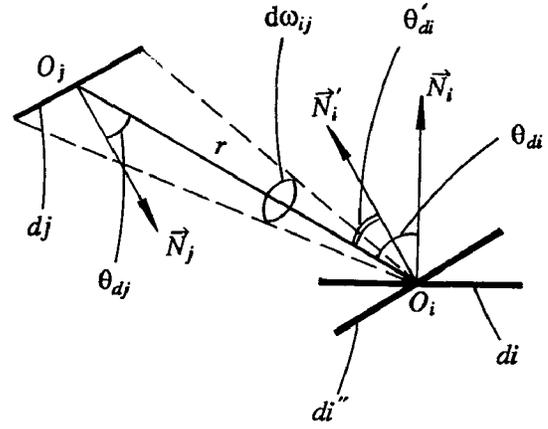


Fig. 5 Rotated Element

As shown in Fig. 5, by rotating the differential element di onto the plane $PL(di, \vec{N}')$, a new differential element di'' is obtained, with the area $dA_i'' = dA_i$. The form-factor from

dA_j to dA_i'' is

$$F_{dj-di}'' = \frac{\cos\theta_{dj} \cos\theta_{di}'}{\pi r^2} dA_i$$

So,

$$F_{dj-di}'' = \frac{\cos\theta_{di}'}{\cos\theta_{di}} \frac{\cos\theta_{dj} \cos\theta_{di}}{\pi r^2} dA_i$$

$$F_{dj-di}'' = \frac{\cos\theta_{di}'}{\cos\theta_{di}} F_{dj-di}$$

Therefore,

$$F_{ji}'' = \frac{1}{A_j} \iint_{\lambda_i} \frac{\cos\theta_{di}'}{\cos\theta_{di}} F_{dj-di} dA_j \quad <20>$$

Similarly, we can get

$$F_{ij}'' = \frac{1}{A_i} \iint_{\lambda_j} \frac{\cos\theta_{di}'}{\cos\theta_{di}} F_{di-dj} dA_i$$

$$A_j F_{ji}'' = A_i F_{ij}''$$

According to the assumption of P-patch i' in previous section, because the distance between the two patches is usually large compared to their sizes, F_{ij}'' is a good approximation to F_{ij}' (the form factor from P-patch i' to shooting patch j). So, equation <9> can be reformulated as

$$H_{ji}' = H_{ji} \frac{F_{ij}''}{F_{ij}} = H_{ji} \frac{F_{ji}''}{F_{ji}} \quad <21>$$

From <10><21>, we have

$$B_i' = E_i + k_i \rho_i \pi \sum_{j=1}^N H_{ji} \frac{F_{ji}''}{F_{ji}} \quad <22>$$

This equation has the desirable characteristics for progressive radiosity design. At each refinement step, the shooting patch j selected may shoot its radiosity to other patches including perturbed patches, say i in $\langle 22 \rangle$. F_{ji}'' and F_{ji} are evaluated at same time by the hemi-cube of shooting patch j .

By our intention of making use of hardware-assisted hemi-cube technique for non-diffuse environment, F_{ji}'' is evaluated numerically as:

$$F_{ji}'' = \sum_{q \in Q} \Delta F_q \alpha_q(i)$$

where Q is the set of hemi-cube pixels covered by the projection of patch i ;

ΔF_q is the standard form-factor related to pixel q .

$\alpha_q(i)$ is $\frac{\cos\theta_i'(q)}{\cos\theta_i(q)}$ if $\cos\theta_i'(q) > 0$, or 0 otherwise;

here $\theta_i'(q)$, $\theta_i(q)$ are the angles between the normal \vec{N}_i' , \vec{N}_i respectively and the direction connecting the center of patch j with the center of pixel q .

For the progressive radiosity method described in Section 2.1, we have implemented both $\langle 10 \rangle$ as a postprocess and $\langle 22 \rangle$ as refinement processing for bump texture generation.

Fig. 4c shows the result of texture generation embodied in the progressive radiosity solution using the latter method, performing the texture operation at each iterative shooting step. The visual difference between Fig. 4b and Fig. 4c is hardly perceptible. It took 113 seconds with 60 iterative steps.

Apparently, the latter method is particularly suitable for texture generation in a radiosity refinement procedure. It allows users to have interactive control of texture generation in the go-between steps. However, when the interactive generation of texture is unnecessary, the former method by texture post-process is preferable to the latter one, because it takes a constant time to produce texture at the end.

2.4 Sampling the Perturbed Normal Vector Space and Rendering

All possible perturbed normal vectors form a hemi-sphere space. In order to reduce the calculation and storage cost, we discretize the hemi-sphere space of perturbed directions. In every sampled direction \vec{N}_s' , the perturbed radiosity $B(\vec{N}_s', \vec{P}_v)$ at a vertex \vec{P}_v can be determined by the algorithm previously mentioned, and stored in a perturbed radiosity map indexed by \vec{N}_s' . It is often sufficient to sample \vec{N}_s' in 16 directions corresponding to E, NE, N, NW, W, SW, S and SE for sun angle 0° and 45° . The more the sample directions, the more accurate the bumpy texture is. For all the colour

figures shown in the paper, 16 directions were utilized.

While rendering the image, if the perturbed normal \vec{N}' at a point is equal to an index \vec{N}_s' of the perturbed radiosity map, its perturbed radiosity can be evaluated by a bilinear interpolation from the radiosities $B(\vec{N}_s', \vec{P}_v)$ at patch vertices. Otherwise, an interpolation of the perturbed radiosities $B(\vec{N}', \vec{P}_v)$ from the adjacent sampled perturbed directions has to be performed at each vertex before the bilinear interpolation is carried out.

Such shading scheme is view-independent for diffuse environment. Besides, *the perturbed radiosity map is independent of perturbation functions*. We may see in the test example a different texture was produced in Fig. 4e using the same perturbed radiosity map as the one in Fig. 4c. An interesting result is shown in Fig. 4f, where the \vec{N}' came from a cylinder function and a cylinder patch was generated by our method combined with displacement map technique[11]. This fact indicates a potential application of radiosity in the process of interactive control of curve simulation during an early stage of shape design. For a non-diffuse textured surface, the specular component of illumination is determined by ray tracing postprocess.

Recently, Wallace et al.[21] and Baum et al.[1] provided two new methods for evaluating form-factors that eliminate the aliasing of hemi-cube algorithm. The true surface normal of receiving element is used in their form-factor algorithms for diffuse environment. Therefore, it is possible for the perturbed radiosity map proposed in the paper to be established using their methods for calculating perturbed form-factors. For example, in Wallace's methodology, the perturbed form-factor from a source 2 to a perturbed differential area dA_1' (Fig. 6) is:

$$dF_{A_2-dA_1}' = dA_1' \frac{1}{n} \sum_{i=1}^n \delta_i \frac{\cos\theta_{1i}' \cos\theta_{2i}}{\pi r_i^2 + A_2/n}$$

where n is the number of sample points on source 2

δ_i is 1 if sample point is visible to vertex 1, 0 if occluded

The perturbed radiosity is obtained :

$$B_1' = \rho_1 B_2 A_2 \frac{1}{n} \sum_{i=1}^n \delta_i \frac{\cos\theta_{1i}' \cos\theta_{2i}}{\pi r_i^2 + A_2/n}$$

Above equation can be evaluated by ray tracing algorithm, and then the perturbed radiosity map at vertex 1 is established. However, the method by hemi-cube is advantageous in two aspects. Firstly, as already shown in the paper, the method for texture is able to be naturally incorporated into our radiosity solution for *non-diffuse* environment while the problem for the other methods in non-diffuse environment is yet to be solved. In fact, introduction of specular

irradiation(H_{ij}^f) derived from hemi-cube has made it possible to produce the texture accounting for specular reflection. Secondly, the hemi-cube algorithm is suitable for hardware implementation, and has high efficiency comparatively.

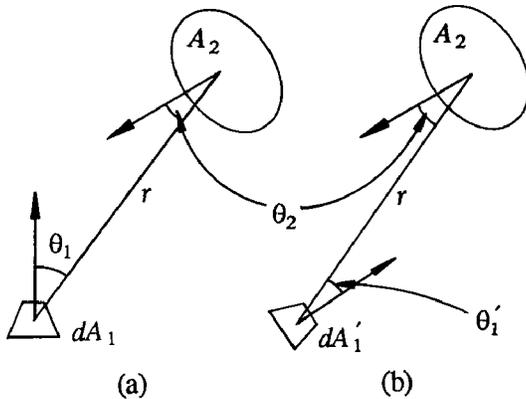


Fig. 6 Geometry for a) Form-factor, b) Perturbed Form-factor in Wallace's Methodology

3. Further Discussion

For different perturbation functions, the distribution of radiant light energy from perturbed patches is different. Fortunately, the difference is usually very slight and can be neglected. Therefore, it is reasonable to take the distribution of non-perturbed radiant energy of patches in an environment as an approximation of that of perturbed patches when calculating the contribution of radiosity from a perturbed patch to the radiosity of other patches in the environment.

When considering so-called *self-shadowing*[14], the bump cast of a part of a surface on nearby parts of the same surface, the "horizon mapping" technique is able to be improved by the newly introduced perturbed hemi-cube. Unfortunately, the processing of self-shadowing will lead the cost of texturing computation to be increased by ninety percent or more. We notice that the effect of self-shadowing in an interreflection environment is often very weak, especially in the case of surface light sources. In order to save computation and storage expense, the present algorithm has not taken the self-shadowing into account like most of the other bump mapping approaches[3][5][15].

Bumps may be illuminated by nearby parts of the surface itself due to perturbation. This phenomenon is referred to as *self-illumination*, in which case, perturbed form-factor F_{ii}' is bigger than zero. If we build a global cube[13] perturbed around patch i , the index of visible patches reserved in the most of the pixels center-symmetrized with the pixels in the shadowed area (Fig. 3) is i , the index of the perturbed patch itself, as the bumpy height is very small. Therefore, we can denote a value to the self-illuminated form-factor F_{ii}'

according to the size of the shadowed area. We get $\sum_{j=1}^N F_{ij}' \leq 1$. The summation of perturbed diffuse form-factors is still not bigger than one. This technique has been used for bump texturing in Figure 4d. Because perturbation is very small, the self-illumination may be neglected.

4. Results

In Fig. 7a and b, a room has been rendered to show the results in a non-diffuse environment. A very bright small source is close to the nearly perfect mirror ($k_d = 0.07$, $k_s = 0.93$). The model contains 578 patches and 3327 elements including 229 perturbed elements and 194 non-diffuse elements. The image on the mirror and the fuzzy image on the sphere ($k_d = 0.25$, $k_s = 0.75$) were produced in the second pass of ray tracing based on the radiosity result from the first pass. The resolution of hemi-cube is 150 by 150. The chessboard-like pattern on the floor and the painting on the wall were generated by color texture mapping technique[9]. The image in Fig. 7a was generated by 30 iterative steps. The texture on the right wall, produced using a random perturbation function looks graceful and the texture on the box is also perceptible due to the specular reflection. Fig. 7b was rendered with 250 iterative steps in 14'31" and the effect of specular reflection is more significant. The bump texture on the box surface has been improved apparently due to more perfect interreflection produced in comparison with Fig. 7a.

In Fig. 8, there are some thinner objects. The scene contains 832 patches and 5776 elements, with 122 perturbed elements for the texture of the light blue curtain sampled at 16 directions, and 64 specular elements for the mirror. A finer resolution hemi-cube, 170 by 170, was utilized. The figure was displayed after 200 solution steps in 12'52". Because the initial patch mesh on the mirror is less fine than in Fig. 7, there is slight hemi-cube aliasing within the region of specular reflection on the right-side wall.

We have also tested that, when images in Fig.7b and Fig.8 are produced without texture generation under the same conditions, their running time is 12'20" (Fig.7b) and 11'23" (Fig.8) respectively. So the net time for texture generation is 2'11" and 1'29", or about 15% and 12% of the total time respectively. The perturbed radiosity maps in Fig.7 and Fig.8 are calculated at each iterative step in the progressive refinement radiosity method.

All figures were measured on SGI Personal IRIS 4D/20 workstation, with hardware-assisted implementation of hemi-cube algorithm.

5. Conclusion

In sum, a bump mapping technique, by the perturbed radiosity map has been proposed in radiosity solution for texture generation in non-diffuse environment. Separate algorithms have been provided for evaluation of the perturbed form-factors to fit different radiosity methods. Consequently, the texture processing is taken either as a postprocess or as a refinement step, incorporated into the procedure of progressive radiosity solution.

The shading scheme based on the perturbed radiosity map is view-independent, and the perturbed radiosity map is also independent of the texturing perturbation function adopted. The bump mapping technique has improved the visual realism of radiosity image and promoted the practical value of radiosity solution.

Acknowledgements

Thanks are given to Yun-Mei Dong and Kei-De Li for providing Apollo-DN580 on which the software was developed during the first stage, and for providing printing facilities. Acknowledgements are also due to Kong-Shi Xu, You-La Zhang and Yu-Guo Wang for their support and various help during the working period. We are grateful to You-Dong Liang and Qun-Sheng Peng of Zhejiang University for their generous help in providing related theses for references long before their papers appeared at conferences. Thanks are also given to SIGGRAPH reviewers for their helpful comments.

The work has been supported by the National Natural Science Foundation.

References

1. Baum, D. R., Rushmeier, H. E. and Winget, J. M., *Improving Radiosity Solutions through the Use of Analytically Determined Form-Factors*, Computer Graphics, Vol 23(3), 1989, pp.325-334
2. Blinn, J. F. and Newell, M. E., *Texture and Reflection in Computer Generated Images*, Comm. ACM, Vol 19(10), 1976, pp.542-547
3. Blinn, J. F., *Simulation of Wrinkled Surfaces*, Computer Graphics, Vol 12(3), 1978, pp.286-292
4. Cabral, B., Max, N. and Springmeyer R., *Bidirectional Reflection Functions from Surface Bump Maps*, Computer graphics, Vol 19(4), 1987, pp.273-281
5. Carey, R. J. and Greenberg, D. P., *Texture for Realistic Image Synthesis*, Computer & Graphics, vol 9(2), 1985, pp.125-138
6. Catmull, E., *A Subdivision Algorithm for Computer Display of Curved Surfaces*, Ph.d Dissertation, Univ. of Uta, Salt Lake City, 1974
7. Chen, H. and Wu, E. H., *An Adapted Solution of Progressive Radiosity and Ray Tracing Methods for Non-diffuse Environment*, Proceedings of CGI'90, June 26-30, 1990
8. Cohen, M. F. and Greenberg, D. P., *The Hemi-cube: A Radiosity Solution for Complex Environment*, Computer Graphics, Vol 19(3), 1985, pp.31-40
9. Cohen, M. F., Greenberg, D. P., Immel, D. S. and Brack, P. J., *An Efficient Radiosity Approach for Realistic Image Synthesis*, IEEE CG&A, Vol 6(2), 1986, pp.26-35
10. Cohen, M. F., Chen, S. E., Wallace, J. K. and Greenberg, D. P., *A Progressive Refinement Approach to Fast Radiosity Image Generation*, Computer Graphics, Vol 22(4), 1988, pp.75-84
11. Cook, R. L., *Shade Trees*, Computer Graphics, Vol 18(3), 1984, pp.223-231
12. Goral, C. M., Torrance, K. E. and Greenberg, D. P., *Modelling the Interaction of Light between Diffuse Surfaces*, Computer Graphics, Vol 18(3), 1984, pp.213-222
13. Immel, D. S., Cohen, M. F. and Greenberg, D. P., *A Radiosity Method for Non-diffuse Environment*, Computer Graphics, Vol 20(4), 1986, 133-142
14. Max, N. L., *Shadows for Bump-mapped Surfaces*, Advanced Computer Graphics, Kunii T.L. Ed. Springer Verlag, Tokyo, 1986, pp.145-156
15. Perlin, K., *An Image Synthesizer*, Computer Graphics, Vol 19(3), 1985, pp.287-296
16. Shao, M. Z., Peng, Q. S. and Liang, Y. D., *A New Radiosity Approach by Procedural Refinements for Realistic Image Synthesis*, Computer Graphics, Vol 22(4), 1988, pp.93-101
17. Shao, P. P., Peng, Q. S. and Liang, Y. D., *Form-factors for General Environments*, Proc. Eurographics'88, Nice, 1988, pp.499-510
18. Sillion, F. and Puech, C., *A General Two-pass Method Integrating Specular and Diffuse Reflection*, Computer Graphics, Vol 23(3), 1989, pp.335-344
19. Sparrow, E. M. and Cess, R. D., *Radiation Heat Transfer*, McGraw-Hill, New York, 1978
20. 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, Vol 21(4), 1987, pp.311-320
21. Wallace, J. R., Elmquist, K. A. and Haines, E. A., *A Ray Tracing Algorithm for Progressive Radiosity*, Computer Graphics, Vol 23(3), 1989, pp.315-324
22. Ward, G. J., Rubinstein, F. M. and Clear, R. D., *A Ray Tracing Solution for Diffuse Interreflection*,



Computer Graphics, Vol 22(4), 1988, pp.85-92

23. Whitted, T., *An Improved Illumination Model for Shaded Display*, Comm. ACM, Vol 23(6), 1980, pp.343-349

(a) (b)

(c) (d) (e) (f)

Figure 4

(a)

Figure 8

(b)
Figure 7