EUROGRAPHICS '93 / R. J. Hubbold and R. Juan (Guest Editors), Blackwell Publishers
© Eurographics Association, 1993

Volume 12, (1993), number 3

A New Radiosity Approach Using Area Sampling for Parametric Patches

Tomoyuki Nishita
Fukuyama University
Higashimura-cho, Fukuyama, 729-02 Japan

Eihachiro Nakamae
Hiroshima Prefectural University
Nanatsuka-cho, Shoubara City, 727 Japan

ABSTRACT

A high precision illumination model is indispensable for lighting simulation and realistic image synthesis. For the purpose of improving realism, research on global illumination has been done, and several papers on radiosity methods have been presented. In the most recently proposed methods, the shapes of light sources and objects are restricted to polygons or simple curved surfaces. We present a more general method which can handle the kind of free-form surfaces widely used in industrial products and in architecture. The method proposed here solves the problem of the interreflection of light (i.e., radiosities) between patches, and form-factors, which play an important role in this process, are precisely calculated without aliasing through the use of an area sampling method (i.e., *pyramid tracing*). Furthermore the method can handle both non-uniform intensity curved sources and non-diffuse surfaces.

Keywords : Radiosity, Interreflection of light, Form-factor, Bézier Surfaces, Scan line algorithm, Shadows, Penumbra

1 Introduction

Radiosity methods have been regarded as an important approach in the generation of realistic images or lighting simulation. In general the following functions for illumination models are required: (a) the precise calculation of illumination (including shadows), (b) light-source geometry (e.g., free-form sources) considering intensity distribution. (c) reflection from diffuse and/or non-diffuse surfaces, (d) the rendering of precise shapes (especially for curved surfaces). In order to satisfy all of the above this paper proposes a more general approach for parametric surfaces.

Radiosity methods are classified into two categories; one, the method solving simultaneous equations relating to finite patches after subdivision of the environment, and the other, that which expands ray tracing (e.g., the rendering equation[11]). In the former, curved surfaces are approximated by using small polygons (in this paper we refer to a subdivided polygon element); both silhouettes of objects and intensities on surfaces are not smooth. In the latter method, the curved surfaces are restricted to simple shapes to allow for the application of the ray tracing method; calculation of additional sampling points for obtaining higher precision is expensive because of the point sampling process.

The method being proposed here belongs to the former category and is especially applicable to parametric patches; surface patches are subdivided, but in such a way that they still maintain a curved surface and so an accurate calculation is guaranteed.

In radiosity solutions using subdivisions, the form-factor is the key element. In this method the hemi-cube method[5] has been most commonly used[9],[6],[18]. Essentially, the form-factor is determined by the fraction of the circle (which is the base of the hemisphere covering a calculation point) covered by projecting the elements onto the hemisphere and then orthographically down onto the circle. The hemi-cube method approximates a hemi-sphere by using a hemi-cube. Regarding the hemi-cube method, some papers [1][23][19] recently pointed out the problems concerning aliasing and accuracy . These problems result both from point sampling and from the approximation to a hemi-cube instead of to a hemi-sphere.

The algorithm proposed here has the following advantages: (1) To improve precision of the form-factor between free-form surface patches, the contour integration method is employed. The algorithm guarantees the required precision of the form-factor by adaptively subdividing an element based on its solid angle viewed from the calculation point. (2) In order to take into account shadow effect, the shadowed ratio of the subdivided element is calculated by using the pyramid composed by the sub-element and the calculation point (we refer this hidden surface algorithm as *pyramid tracing*). The intersection test between the pyramid and surface patches employs an extended version of *Bézier Clipping* which is developed for ray tracing[16]. The aliasing problem is reduced because this algorithm employs area sampling instead of point sampling used in previous methods. (3) For the radiosity of the non-diffuse component, the two pass solution is employed. The precision is improved by taking account of solid angle of the element. (4) In previous methods, light-source geometry is restricted to polygons or polyhedra. In the real world, most sources are composed of curved surfaces such as bulb lamps. The intensity distribution on the surfaces of light sources are usually non-uniform. The lighting model proposed here can handle curved surface sources of non-uniform luminance. As sky light is considered to be a large hemisphere with non-uniform intensities, the proposed algorithm is useful for rendering not only indoor but also outdoor scenes.

2 Previous Work

In this section, the calculation methods of form-factors playing the most significant part in radiosity solutions are reviewed. Since Cohen et al. first presented the hemi-cube method[5], it has been employed in many papers. This method is useful not only for the calculation of form-factors but also for shadow detection. The hemi-cube method basically is regarded as the z-buffer algorithm when the center of an element is assumed as being the viewpoint. It is here that the problems of aliasing and precision arise because of the point sampling employed in the z-buffer algorithm. The authors previously calculated form-factors which takes into account shadows by using visible segments of the boundaries of elements arrayed sequentially, assuming those boundaries as linear light sources[13]. Though this approach is more accurate than point sampling approaches by virtue of the fact that it is a line sampling, this method is restricted to convex polyhedra.

To improve the precision of form-factors, Baum [1] solved form-factors analytically, instead of using the hemi-cube method, but only for elements where large errors occur (e.g., between adjacent elements).

The hemi-cube method is also regarded as hidden surface removal using rays firing from the center of an element (a calculation point); the elements to be detected sometimes exist between rays, and the aliasing problem occurs because of sampling miss. Wallace et al. solved this problem by firing the rays from the elements to be detected to the calculation point[23]. Since the number of rays fired from each element is very few (or only one) in their examples, this method, however, still lacks precision. In other words, it is difficult to arrive at a precise solution with this method because of its use of point sampling. Sillion [20] also employed this method.

In the hemi-cube method, the five surfaces on a hemi-cube other than the base surface are considered as screens; sampling is then done at each pixel on the screens. Sillion [19] used only one large screen corresponding to the top surface of a large hemi-cube, and employed hidden-surface removal using area coherence, such as Warnock's algorithm[24], on the screen after projecting the elements onto it. This method can be regarded as a kind of area sampling; the screen is subdivided into a variable-size mesh (each subdivided cell is called a proxel), each proxel contributes about the same amount to the form-factor. The form-factor for an element is calculated by integrating every proxel within the visible parts of the element. However, the errors in calculation cannot be avoided since the mesh does not have radial symmetry, as the authors themselves point out. Recently Baranoski[2] developed the parametric differential method of form-factors for curved surfaces. However, the method is applied only for spheres, and the occlusion test is performed by point sampling.

Radiosity solution for non-diffuse surfaces was first developed by Immel[9]. This method solves simultaneous equations by taking into account discrete directions from each element. One of the drawbacks of this method is that the array size requires the square of the number of elements to be multiplied by the number of discrete directions. Shao et al.[18] developed a new radiosity algorithm for non-diffuse environments by procedual refinement so that the calculation cost for specular reflection is reduced. However, the method still requires a large memory capacity, and there is also the problem of aliasing. Wallace et al.[22] proposed a two pass solution by which a combination of ray tracing and radiosity technique is adopted. Sillion et al. [19] extend Wallace's two pass solution and the aliasing problem is reduced. Most of the methods mentioned above do not deal with parametric surfaces. The algorithm proposed here can deal with non-diffuse parametric surfaces. Even though the algorithm also employs a two pass solution, precision is improved by taking into account the solid angle of the element.

3 Basic Ideas

Bicubic Bézier surfaces are used in this paper because almost all surfaces can be converted into these. Curved surfaces are subdivided into sub-patches. In this paper we refer to an original surface as a patch, and a sub-patch as an element. To calculate form-factors, elements are adaptively subdivided; these subdivided elements are referred to as *sub-elements*. No element has polygonal approximation but it still maintains a curved surface, and its edges are also curves even though each sub-element is approximated by a polygon. Before discussing on form-factors, the basic idea of hidden surface removal for parametric patches is described because the idea is used to calculate form-factors taking into account shadows. In our method, full-matrix radioaitity is used.

3.1 Hidden surface removal of parametric surfaces

We propose three algorithms to realize precise hidden surface removal of parametric surfaces. Our hidden surface removal is an extension of Lane and Carpenter's algorithm[12]: in their method curved surfaces are subdivided into polygons on each scanline, but small gaps arise between approximated polygons. We solved this problem by subdividing curved surfaces on each scanline into nearly flat subpatches with curved edges, which are rendered by scanning.

For the scanning algorithm, we employ the author's scanline algorithm[17]using Bézier Clipping[16] because of the following advantages: Bézier Clipping is an iterative method which takes advantage of the convex hull property of Bézier curves, provides more robust convergence to the solution than Newton's method, and can be used to find a solution for curve/plane, surface/plane, and ray/surface intersections.

As a post-processing step, a ray tracing algorithm[16] suitable for Bézier patches is used to display reflected objects.

3.2 Form-factors

Form-factors are generally obtained by summing/integrating differential form-factors (i.e. a point-element form-factor or an area-to-differential-area form-factor). We discuss here on differential form-factors. Since form-factors depend on the apparent area (solid angles) of an element viewed from a calculation point (i.e., the center of a differential area), the boundary shapes of elements affect the precision of form-factors. In cases where boundaries are straight lines, it is well known that the contour

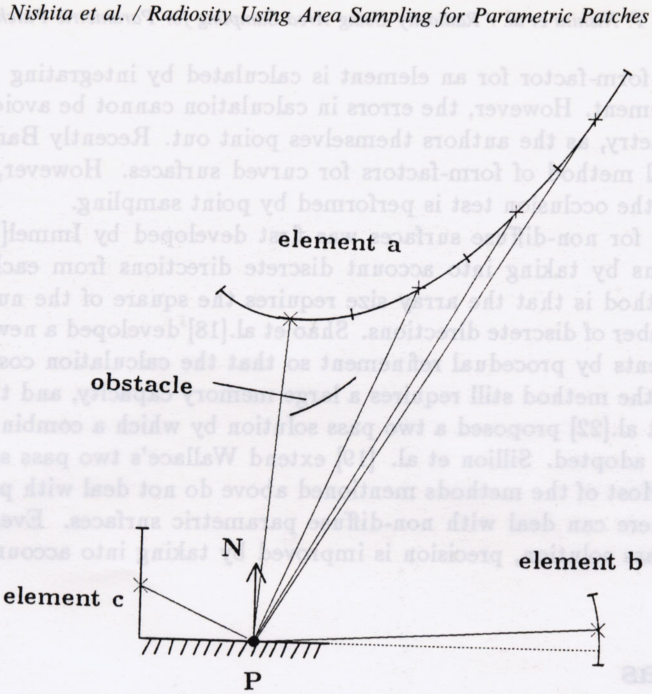


Figure 1: Aliasing problems in point sampling algorithm.

integration method can be used to get an analytical solution of form-factors, which will be described in 4.2 in detail. The authors used the contour integration method for calculating direct illumination due to area sources[15]. Baum et al. [1] also partially used it for solving form-factors. This method is very useful when the boundaries are straight lines, while it is difficult to obtain any analytical solution of form-factor between the surfaces with curved boundaries.

Most algorithms have approximated free-form curved surfaces by polygons, but this gives rise to a number of sticky problems with form-factors. When a pair of surfaces is located very close together, the form factor becomes large, and use of polygonally approximated surfaces can introduce a large error in the form-factor calculation. Further, if surfaces are subdivided precisely, calculation of form-factors for distant surfaces is expensive so an adaptive polygonal approximation is desirable. The error in surface normal vectors introduced by polygonal approximation also aggravates the calculation error. The method presented here solves these problems by using area sampling based on *pyramid tracing* which is discussed in section 4.

For form-factor calculations which take shadows into account, hidden surface removal is a very important processing step; it is complex when free-form curved surfaces are involved. To take into account shadows, visible segments of the boundary viewed from a calculation point must be detected; it is difficult to guarantee adequate precision in the case of free-form surfaces. Among methods which avoid polygonal approximation are those based on Z-buffer[7] and ray tracing[10]. These methods, however, perform computationally expensive ray/surface intersection. To overcome this problem our method applies the Bézier clipping method described in section 4.2: We use a hidden surface algorithm for polygons after adaptively subdividing elements into subelements until adequate precision can be guaranteed: uniform error distributions which user expect can be realized not like the following previous method.

Actually, our proposed method is similar to Wallace's [23], but his method has the following problems (Fig.1 shows the relationship between calculation point P and some source elements): (1) There is no sampling miss for source elements, the algorithm leaves the possibility of missing obstacles casting shadows (see element a in Fig.1). Even though the sampling point on an element exists in front of the plane which contains the calculation point, some parts of the element are hidden by the plane (see element b in Fig.1); in this case the error is relatively large. (2) The error arises due to approximation

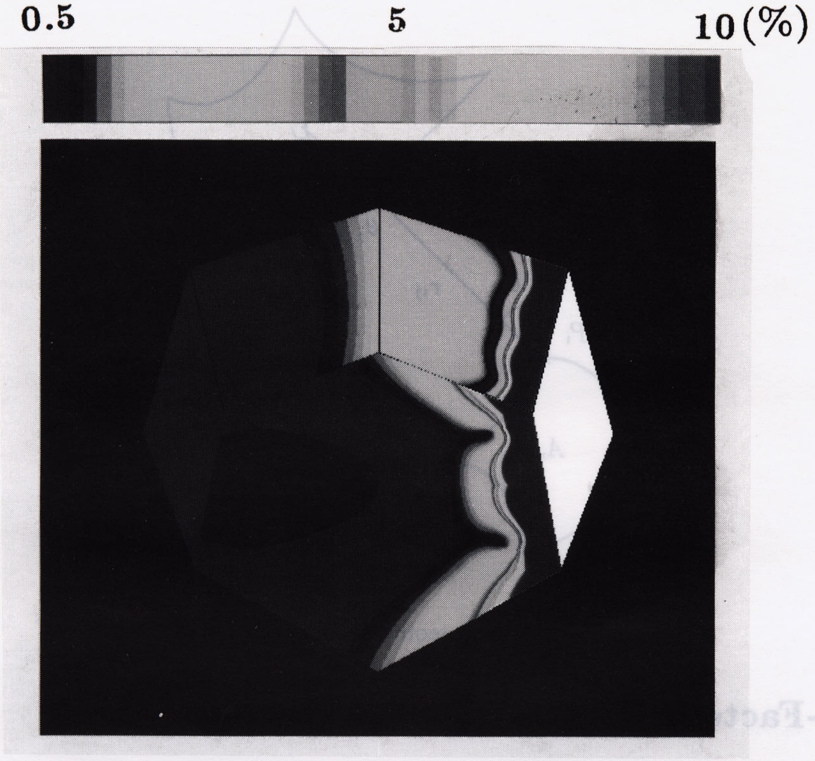


Figure 2: Example of the error distribution of form-factor (point-element form-factor) approximated by disks (Perspective of a hexagonal box with a rectangular source).

of an element by disks. When a pair of elements are located very close together or the angle between the normal vector of the element and the normal vector at the calculation point is large (see element c in Fig.1), they result in a large error in the form-factor calculation. As Baum [1] has pointed out, aliasing between adjacent faces, such as the corner edge of a wall and a floor is significant in most of the previous methods, and Wallace's method is no exception. Fig.2 shows one of the examples of the error distribution of point-element form-factor calculated by his algorithm; the error distribution on a hexagonal floor and three rectangular walls are depicted when the source element with white rectangular is subdivided 3 by 3. For the distant faces (blue rectangular; left side of figure), the error is very small (less than 0.2%), but for the corner of the surface large errors (more than 40%) exist. In our method the error is 0.05 % on every calculation point.

Further, for each ray in his method its differential area (approximated by a disk) is required, but the calculation of the area of a curved surface is difficult, and it is necessary to calculate the surface normal at each point on the surface. Our method requires neither area nor surface normal of each source element.

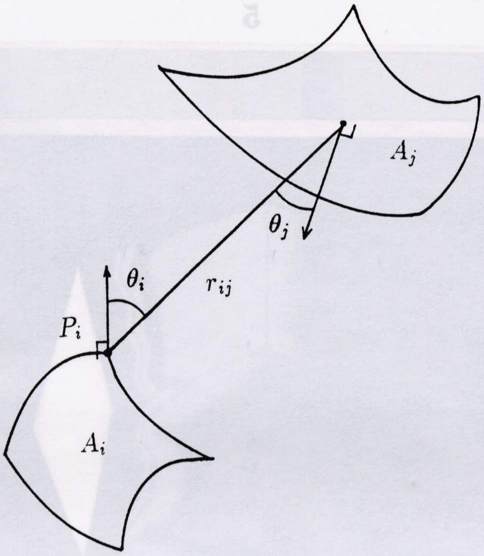


Figure 3: Form-factor geometry between two elements.

4 Form-Factors Using Pyramid Tracing

4.1 Radiosity equation

Assuming the form-factor from element i of area A_i to element j of area A_j to be F_{ij} (see Fig.3), as it is well known, radiance E_i at element i is given by the following;

$$E_i = E_{0i} + \rho_i \sum_{j=0}^{n} F_{ij} E_j, \tag{1}$$

where ρ is reflectance, E_{0i} direct radiance, E_j radiance at element j, and n the number of elements. The radiance at each element can be obtained by solving the n-dimension of simultaneous equations derived by eq. (1).

The form-factor from a differential element dA_i (surrounding point P_i in Fig.3) to element j is given by,

$$F_{dA_iA_j} = \frac{1}{\pi} \int_{A_j} \frac{cos\theta_i cos\theta_j}{r_{ij}^2} dA_j. \tag{2}$$

The form-factor, which means the fraction of energy from sending element i to receiving element j, is obtained by area-averaging:

$$F_{A_iA_j} = \frac{1}{\pi A_i} \int_{A_i} \int_{A_j} \frac{cos\theta_i cos\theta_j}{r_{ij}^2} dA_j dA_i. \tag{3}$$

Now, the radiance due to area sources at a vertex is calculated by the use of form-factors. The radiance at a point on element i is calculated by multiplying $F_{dA_iA_j}$, for the point, with the reflectance of element i, where element j is regarded as an area source. Intensities are usually calculated by using $F_{A_iA_j}$; the radiance is calculated at the center of a receiving element, for example, in the hemi-cube method. In previous methods, as F_{ji} is calculated by using the relationship, $F_{ji} = A_i/A_jF_{ij}$, the areas, A_j, A_i, are required.

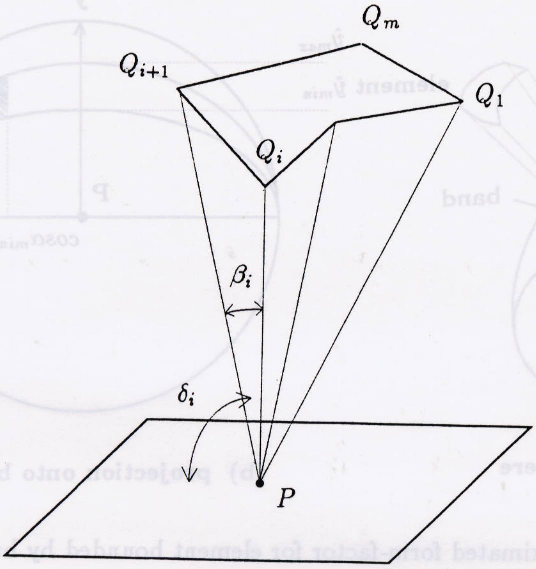


Figure 4: Contour integration for area source.

4.2 Proposed algorithm

The area of each element is necessary for the calculations of the form-factor in eq. (3). The areas are also necessary for the conversion from $F_{dA_iA_j}$ to $F_{dA_jA_i}$.It is, however, difficult to get precise areas because of free-form surfaces. In the proposed method using $F_{dA_iA_j}$, the area of any receiving element is not required; radiance is calculated at every corner of the receiving element.

Let's discuss how to calculate form-factors using the contour integration used for polygonal sources. If the source is the m-sided polygon, calculation is simple; the following contour integral can be employed (see Fig.4).

$$I = \frac{1}{2}\sum_{i=1}^{m} \beta_i \cos\delta_i, \tag{4}$$

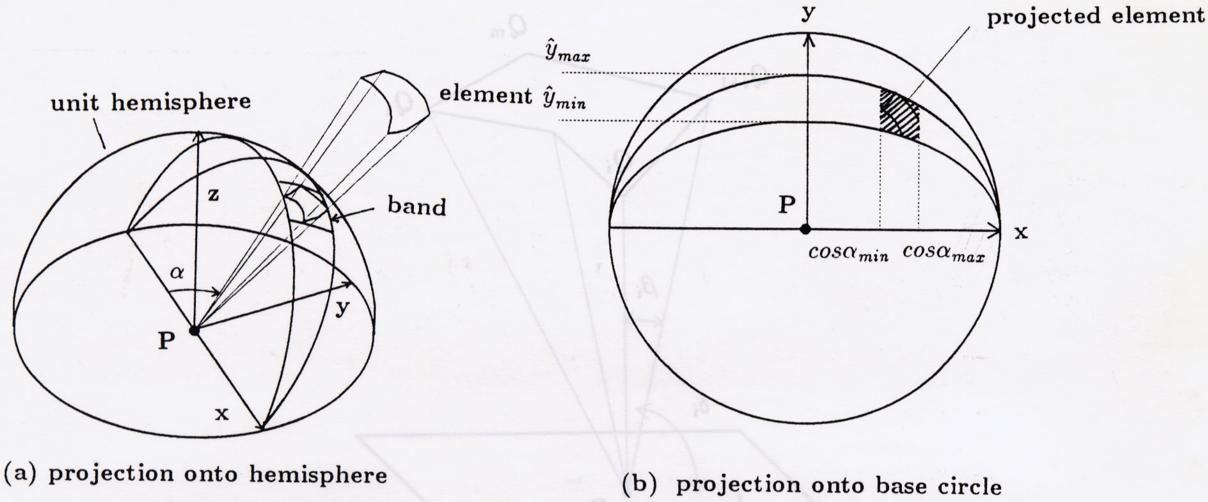
here β_i is the angle between vectors PQ_i and PQ_{i+1}, and δ_i is the angle between the normal of triangle PQ_iQ_{i+1} and the normal of the plane including point P.

In order to apply equation (4)to curved surfaces, the following polygonal approximation is required. Essentially, the form-factor is determined by the fraction of the circle covered by projecting the elements onto the base of the hemisphere. Then the number of subdivision of boundary curves into line segments is determined by the flatness test[12] using control points of a Bézier patch on the base circle after projection. When shadow effects are considered, the subdivision of boundary curves is not sufficient. To simplify the calculation, the following algorithm is used.

If the form-factor of an element is small, the approximation error is ignored. Therefore, after the calculation of the approximated form-factor described in the next subsection, the element is subdivided into sub-elements in order to applying the contour integration for boundaries of the subelements.

4.2.1 Approximated form-factor for element

As shown in Fig.5, we set the bounded area of a projected element on the base circle. The form-factor is calculated by employing this area. This is equivalent to the projected area of band source used for sky light [14]. The band is determined by two planes, which contain x-axis and bound every control points of the element: the interval of the band is obtained by the minimum and maximum angles (α_{min} and α_{max}) of control points from the x-axis. The form-factor (i.e., the projected area of



(a) projection onto hemisphere (b) projection onto base circle

Figure 5: Approximated form-factor for element bounded by band element.

the band) is derived by the following equation;

$$F = (\hat{y}_{max} - \hat{y}_{min})(E(\alpha_{min}) - E(\alpha_{max})), \tag{5}$$

where

$$E(\alpha) = (\alpha - cos\alpha sin\alpha)/(2\pi) \quad (0 \leq \alpha \leq \pi),$$

$cos\alpha$ is determined by the inner product between x-axis and the vector from the calculation point to control point P_{ij}, and $\hat{y}_{min}$ and $\hat{y}_{max}$ are determined by $\hat{y}_{ij}$ which is defined by

$$\hat{y}_{ij} = Y_{ij}/\sqrt{Y_{ij}^2 + Z_{ij}^2}, \tag{6}$$

where (X_{ij}, Y_{ij}, Z_{ij}) are control points of the element. Note that when the element is a light source, high precision is required even though the form-factor is very small. Then the number of element subdivision is propotional to the function $s(= F\cdot$ intensity of light). If s is smaller than a given tolerance, we may approximate the form-factor by F (the shadow detection also ignored). As shown in Fig.5(b), it is clear that the true form-factor does not exceed F.

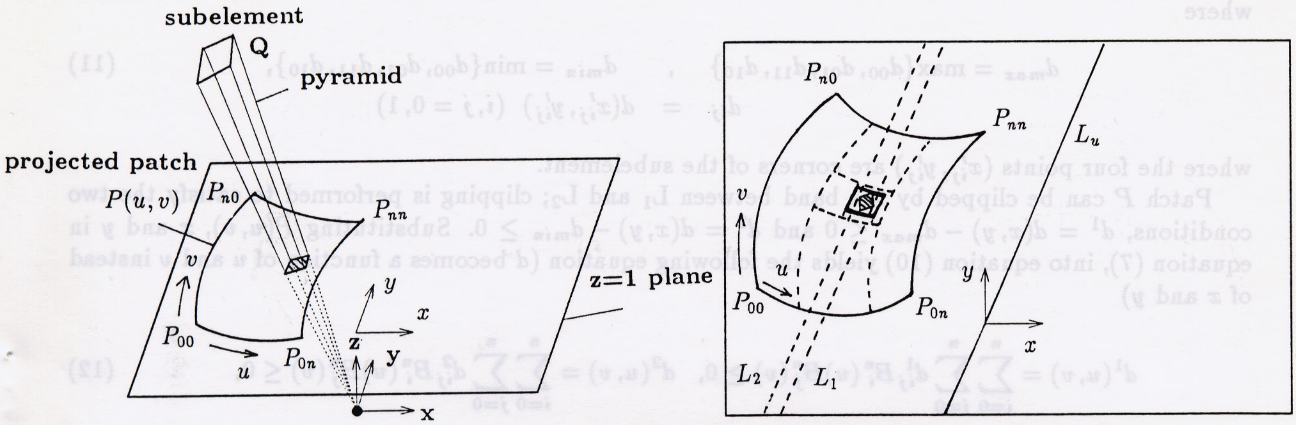
4.2.2 Form-factor calculation taking account of shadows

As mentioned before, to apply contour integration to the calculation of form-factors, visible segments of boundary curves should be detected. In the case of free-form surfaces, the detection of segments is complicated, even though it is easy for polygons. Then each form-factor is calculated by multiplying the form-factor of the subelement mentioned before by the shadow ratio, which is the ratio of visible parts to the whole subelement area; the shadow ratio is obtained by the following method.

Even if a boundary of the subelement is a curved line, the element can be approximated by a 4-sided polygon because the element is subdivided sufficiently small to get an adequately precise form-factor.

The shadow ratio is obtained by the intersection test between the pyramid composed of the calculation point and the 4 corners of the subelement and each surface patch(see Fig.6(b)). This test is performed on a 2-D plane using the Bézier clipping method[16].

It's assumed here that the origin of the coordinate systems is the calculation point, the z-axis coincides with the normal at the calculation point, the x-axis is set in an arbitrary perpendicular



(a) 3-D space

(b) 2-D plane

Figure 6: Pyramid tracing using Bézier Clipping.

direction to the z-axis, and a plane($z = 1$) is the projection plane (see Fig.6(a)). The control points of each Bézier patch are projected onto the projection plane. Let's consider Bézier surface $P(u, v)$ of degree n. The projected patch is calculated by

$$\begin{aligned} x(u,v) &= \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}x_{ij}B_i^n(u)B_j^n(v)}{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}B_i^n(u)B_j^n(v)}, \\ y(u,v) &= \frac{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}y_{ij}B_i^n(u)B_j^n(v)}{\sum_{i=0}^{n}\sum_{j=0}^{n} w_{ij}B_i^n(u)B_j^n(v)}, \end{aligned} \tag{7}$$

where B is the Bernstein polynomial given by $B_i^n(u) = \binom{n}{i}u^i(1-u)^{n-i}$. The homogeneous coordinates (x_{ij}, y_{ij}, w_{ij}) of each control point P_{ij} of a surface on the projection plane are given by

$$x_{ij} = X_{ij}/Z_{ij}, \quad y_{ij} = Y_{ij}/Z_{ij}, \quad w_{ij} = W_{ij}Z_{ij}, \tag{8}$$

where (X_{ij}, Y_{ij}, Z_{ij}) are control points of Bézier Patch, and w_{ij} is a control point weight.

Even though Bézier clipping was developed for ray/surface (or point/surface on 2-D plane) test, we can employ this for the extraction of the overlapped area between the surface patch and a specified region; the interval of parameters, u and v, which overlap the region are extracted, then the patch is clipped by using these intervals.

First, let's discuss the subdivision of component u by considering patch $P(u, v)$ and subelement Q in Fig.6. In the figure the direction of line L_u is the same as the average direction of two edges of the patch, $P_{00}P_{0n}$ and $P_{n0}P_{nn}$, and the line passes through the origin of the projection plane.

The distance d from an arbitrary point (x, y) to line L_u on the projection plane is given by

$$d(x,y) = ax + by + c, \tag{9}$$

where (a, b) is the unit normal of L_u (i.e., $a^2 + b^2 = 1$), and $c = 0$ because L_u passes through the origin.

The projected subelement is enclosed by two lines, L$_1$ and L$_2$ (see Fig.6). Within the bounded region the following relation is satisfied:

$$d_{max} \geq d(x,y) \geq d_{min}, \tag{10}$$

where

$$d_{max} = \max\{d_{00}, d_{01}, d_{11}, d_{10}\} \quad , \quad d_{min} = \min\{d_{00}, d_{01}, d_{11}, d_{10}\}, \qquad (11)$$
$$d_{ij} = d(x'_{ij}, y'_{ij}) \quad (i,j = 0,1)$$

where the four points (x'_{ij}, y'_{ij}) are corners of the subelement.

Patch P can be clipped by the band between L$_1$ and L$_2$; clipping is performed to satisfy the two conditions, $d^1 = d(x,y) - d_{max} \leq 0$ and $d^2 = d(x,y) - d_{min} \geq 0$. Substituting $P(u,v)$, x and y in equation (7), into equation (10) yields the following equation (d becomes a function of u and v instead of x and y)

$$d^1(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} d^1_{ij} B^n_i(u) B^n_j(v) \geq 0, \quad d^2(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} d^2_{ij} B^n_i(u) B^n_j(v) \leq 0, \qquad (12)$$

where

$$d^k_{ij} = w_{ij}(ax_{ij} + by_{ij} + c^k) \quad (k = 1,2), \qquad (13)$$
$$c^1 = c - d_{min}, \quad c^2 = c - d_{max}.$$

Therefore, by using Bézier Clipping it is possible to find the interval of u satisfying the relationship $d_{max} \geq d(u,v) \geq d_{min}$. When some control points exist behind the surface including calculation point P, the points can not be projected, so the following equation in 3-D space is used instead of equation (13).

$$d^k_{ij} = W_{ij}(aX_{ij} + bY_{ij} + c^k Z_{ij}) \quad (k = 1,2). \qquad (14)$$

For the v-component of the surface patch, the same algorithm as mentioned above is used. After extracting the intervals of u and v-components, the surface patch is clipped. As shown in Fig. 6(b) a small patch (solid line) is obtained. It is guaranteed that some regions of the patch overlapping onto the subelement will always remain. If the intervals are not small enough, the surface patch is subdivided recursively. If equation (12) is not satisfied, the patch does not overlap; this test is very simple because of the convex hull property of the Bézier surface; if every control point exist outside L_1 and L_2 (i.e., $d^1_{ij} \leq 0$ for every P_{ij} or $d^2_{ij} \geq 0$ for every P_{ij}), the patch does not overlap.
In order to save calculation time, the bounding box test on the projection plane is performed before the above test; the maximum and minimum of y coordinates are used for equation (6).

By using the test mentioned above differential patches overlapping the subelement are obtained. Since these subpatches are small, they can be approximated by polygons. The visible area of the subelement is calculated by conventional hidden surface removal algorithm for polygons such as the Weiler-Atherton clipping algorithm, scanline algorithm, or Z-buffer algorithm (hardware). In this paper a scanline algorithm is employed.

4.2.3 Specular reflection

In this paper Phong's reflection model is employed. In general point sampling (or ray tracing) is used for the calculation of specular reflection. Here we consider the solid angle of the source element. In Phong's model, specular reflection I_e for the element is expressed by[21]

$$I_e = I_0 r_s \frac{(n+1)}{2\pi} \int_{\Omega} (R \cdot L)^n d\omega, \qquad (15)$$

where R is the reflected vector, L, the light vector, Ω, the solid angle for the element, I_0, the intensity of the source element, r_s, specular reflectance, and n, the degree of sharpness. Even though the analytical method for the specular reflection of area light sources has been developed by Tanaka [21], this method is not applicable to shadow effects. We use the following numerical method; equation (15) is descretized

$$I_e = I_O r_s \frac{(n+1)}{2\pi} \sum_{k=1}^{m} (R \cdot L_k)^n \Delta\omega_k, \tag{16}$$

where $\Delta\omega_k$ is the solid angle for subelement k, and m is the number of subelements. The solid angle of a subelement is obtained by the its form-factor because it is defined by the projected area of the solid angle onto the base circle of unit hemisphere. The solid angle is calculated by dividing the form-factor by $cos\gamma$ angle of which is formed by the direction from the calculation point to the center of the subelement,L_k, and the normal at the point. If the inner product between vector R and the vector L_k is larger than a given tolerance ϵ_1, rays are increased to calculate the term of $(R \cdot L_k)^n$ precisely (L_k is distributed on the subelement). On the other hand, if the inner product is smaller than a given value ϵ_2, the calculation can be ignored. That is, for the elements towards the reflection direction the sampling points are increased, else the specular component can be ignored.

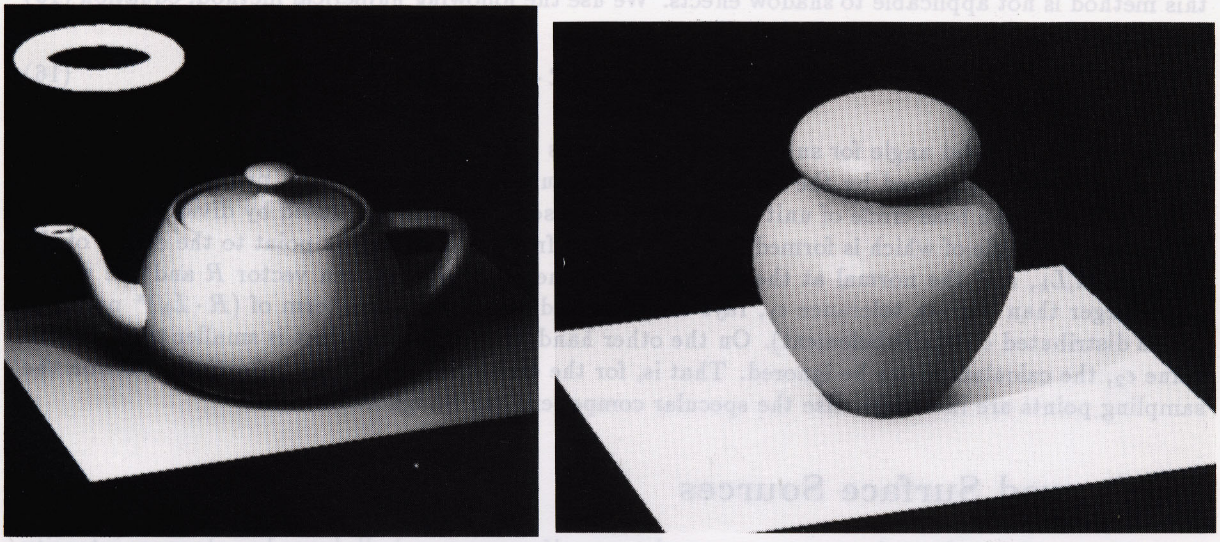
5 Curved Surface Sources

Intensities on the surface of a source are not always uniform; e.g., a bulb lamp has stronger intensity around its illumination axis. Curved sources are dealt with in the same manner as the illuminated elements mentioned in the above section; by subdividing the curved surfaces into small curved elements, the sampling method mentioned above can be applied. The proposed method can also deal with non-uniform intensity sources with perfect diffuser (i.e., Lambertian source) by specifying intensity at every corner point of each element; the intensity of each element is estimated by the average value of intensities at the four corners of the element. Note that a curved source is different from a flat source because of its sometimes cutting the light by itself, which is why the shadow processing described in the previous section is necessary. Sky light is considered to be a large hemisphere called a sky-dome; sky luminance has non-uniform distribution, and the distribution and spectrum depend on the sun position[14]; the algorithm proposed here can be applied to sky light.

6 Patch Subdivision

Cohen et al.[6] developed an adaptive subdivision method in which environments are subdivided into small elements and further subdivision is done for areas with high radiance gradients - a binary subdivision technique splitting an element into four smaller elements. In this method, however, once a curved patch is approximated by polygons, the further subdivided polygons do not always exist on the original surface. Campbell et al.[3] recently developed new adaptive subdivision methods for displaying shadow boundaries, but these methods are limited to polygonal environments. The proposed method also employs adaptive subdivision, but the subdivided elements are still curved surfaces; This means that the algorithm guarantees exact surfaces after further subdivisions. In our method, the de Casteljau Algorithm, which is well known, is used for the subdivision of Bézier patches.

Generally, we find some areas with high gradients of intensity, such as neighbors of shadow boundaries and light sources (the intensity follows the inverse-square law of distance). Since these areas depend on direct light from the sources, the elements to be subdivided are detected after calculation of direct illumination. If the intensities at the four corners of an element are considerably different, the element is subdivided into four sub-elements; this process is similar to the adaptive supersampling used in a ray tracing algorithm. If there are still larger differences between those subdivided elements, the elements are subdivided again.



(a) torus source

(b) sky light

Figure 7: Basic Examples.

7 Rendering

7.1 Hidden surface removal

As mentioned in section 3, hidden-surface removal is done by scan-line algorithm[17]. Patches are subdivided into almost flat sub-patches, the intersections between scanlines and the edges of sub-patches are calculated. Since intersection segments between the scan line and sub-patches are assumed as straight lines (after finding intersection points), the visible segments are obtained by using a conventional scan-line algorithm. One of advantages in the proposed method is that the same technique, Bézier Clipping, used for calculation of the form-factor, can be used.

7.2 Shading at each pixel

In methods developed until now, both direct and indirect intensities at inner points of an element are interpolated by using the intensities of its corners. Since the intensity gradient of a direct component is relatively large especially around shadow boundaries, it is difficult to render fine variation even if the subdivision of elements is considerably dense. To overcome this, direct and indirect components are stored in separate arrays, and then all of the indirect components and the direct components with small gradients of radiosity are interpolated from intensities at the four corners, and only the remaining direct components with large gradient of radiosity are calculated at every few pixels.

8 Examples

Fig. 7 demonstrates two basic examples of the proposed method. Fig. (a) is an example displaying non-diffuse surfaces illuminated by a curved light source; a teapot with a non-diffuse surface is illuminated by a torus source which is a perfect diffuse source; highlights are observed. Fig. (b) is an example of a curved object with a non-diffuse surface illuminated by a curved source with non-uniform intensity; the object is illuminated by sky light (i.e., a large hemi-sphere) with clear sky luminance distribution[15] and direct sunlight. Here the sun is treated as a small area source whose apparent angle is only 16 minutes, even though it is usually dealt with as a parallel light; most previous radiosity approaches may have shadow problems due to very small light sources such as the sun.

Fig. 8*(a) displays the lighting effects of an interior (a room with a curved wall) illuminated by direct and indirect lighting; two curved sources are set on the ceiling and on the left wall, and two hidden sources, a rectangular source and a curved source, are put behind the walls. All curved surfaces in the room, the curved wall and pots, are dealt with as non-diffuse surfaces, while the others are diffuse surfaces. For the mirror the ray tracing method is used. In this figure, 79 Bezier patches(1700 elements) are used, and the cpu time is 18.5 min.(500 x 400 resolutions) by using Iris Indigo Elan(R4000).

Fig.(b) shows a racing car illuminated by two area light sources. The scene is composed by 762 Bezier patches(5440 elements), and the cpu time is 74.7 min.. Fig.(c) shows a lobby (data from Yamagiwa Co. Ltd.) as a practical example. There are 4 light sources, a large curved boundary source (ellipse-like) surrounding by a concave curved surface, a rectangle source in the elevator, two hidden vertical rectangle sources (left side of the curved wall and left side of the elevator). This scene is composed of 81 Bezier patches(1450 elements). The cpu time is 95.2 min.(500 x 400 resolutions). In Fig.(a) and (c), light sources are very close to the ceiling or walls, and lit them; the proposed method is effective for such close light sources.

Fig.(d) shows a conference room illuminated by 24 area light sources. This scene is composed of 919 Bezier patches(7520 elements). The cpu time is 518.7 min..

The method proposed here is useful for lighting pre-evaluation including such complex curved sources and surfaces.

9 Conclusion

A calculation method with accurate radiosities for environment including curved surfaces has been discussed. The advantages are as follows;

(1) The method can calculate radiosity for diffuse and non-diffuse parametric surfaces.

(2) Precise form-factors with anti-aliasing are calculated by using an area sampling(*pyramid tracing*); no sampling miss occurs even for small elements.

(3) The method gives sufficient precision of radiosities and effective adaptive subdivision of elements with high intensity gradients because of non-polygonal approximation.

(4) The method can deal with curved light sources with non-uniform intensity distribution.

References

[1] D.R. Baum, H.E. Rushmeier, J.M. Winget: "Improving Radiosity Solutions Through the Use of Analytically Determined Form-Factors," Computer Graphics, Vol.23, No.3(1989) pp.325-334.

[2] G. Baranoski: "The Parametric differential Method: An Alternative to the Calculation of Form-Factors," Eurographics '92 Vol.11, No.3(1992) pp.193-204

[3] A.T. Campbell, D.S. Fussell: "Adaptive Mesh Generation for Global Diffuse Illumination," Computer Graphics, Vol.24, No.4(1990) pp.155-164.

[4] H. Chen, E. Wu: "An Efficient Radiosity Solution for Bump Texture Generation," Computer Graphics, Vol.24, No.4(1990) pp.125-134.

[5] M.F.Cohen, D.P. Greenberg: "The Hemi-Cube: A Radiosity Solution for Complex Environments," Computer Graphics, Vol.19, No.3(1985) pp.31-40.

[6] M.F.Cohen, D.P.Greenberg, D.S. Immel, P.J.Brock: "An Efficient Radiosity Approach for Realistic Image Synthesis," IEEE CG & A, Vol.6, No.3(1986) pp.26-35.

* See page C–536 for Figure 8.

[7] E.E.Catmull: "Computer Display of Curved Surfaces", Proc. IEEE Conf. Computer Graphics, Pattern Recognition and Data Structures, (1975) p.11.

[8] P.S. Heckbert: "Adaptive Radiosity Textures for Bidirectional Ray Tracing," Computer Graphics, Vol.24, No.4(1990) pp.145-154.

[9] D.S.Immel, M.F.Cohen, D.P.Greenberg: "A Radiosity Method for Non-Diffuse Environments," Computer Graphics, Vol.20, No.4(1986) pp.133-142.

[10] J.T. Kajiya: "Ray Tracing parametric patches," Computer Graphics, Vol.16, No.3(1982) pp.245-254.

[11] J.T. Kajiya: "The Rendering Equation," Computer Graphics, Vol.20, No.4(1986) pp.143-150.

[12] J.M. Lane,L.C. Carpenter T. Whitted J.F.Blinn: "Scan Line Methods for Displaying Parametrically Defined Surfaces," CACM, Vol.23, No.1(1980) pp.23-34.

[13] T. Nishita, E. Nakamae: "Continuous Tone Representation of Three-Dimensional Objects Taking into Account of Shadows and Interreflection," Computer Graphics, Vol.19, No.3(1985) pp.23-30.

[14] T. Nishita, E. Nakamae: "Continuous Tone Representation of Three-Dimensional Objects Illuminated by Sky Light," Computer Graphics, Vol.20, No.3(1986) pp.125-132.

[15] T. Nishita, E. Nakamae: "Half-Tone Representation of Three-Dimensional Objects Illuminated by Area or Polyhedron Sources," IEEE COMPSAC (1986) pp.239-241.

[16] T. Nishita, T. Sederberg, M.Kakimoto: "Ray Tracing of Trimmed Rational Surface Patches," Computer Graphics, Vol.24, No.4 (1990), pp.337-345.

[17] T. Nishita, K.Kaneda, E. Nakamae: "A Scanline Algorithm for Displaying Trimmed Surfaces by using Bezier Clipping," The Visual Computer, Vol.7, No.5(1991) pp.269-279.

[18] M. Shao, Q. Peng, Y. Liang: "A New Radiosity Approach by Procedural Refinements for Realistic Image Synthesis," Computer Graphics, Vol.22, No.4(1988) pp.93-101.

[19] F. Sillion, C.Puech: "A General Two-Pass Method Integrating Specular and Diffuse Reflection," Computer Graphics, Vol.23, No.3(1989) pp.335-344.

[20] F. Sillion, J. Arvo, S. Westin, D. Greenberg: "A Global Illumination Solution for General Reflectance Distributions," Computer Graphics, Vol.25, No.4(1991) pp.187-196.

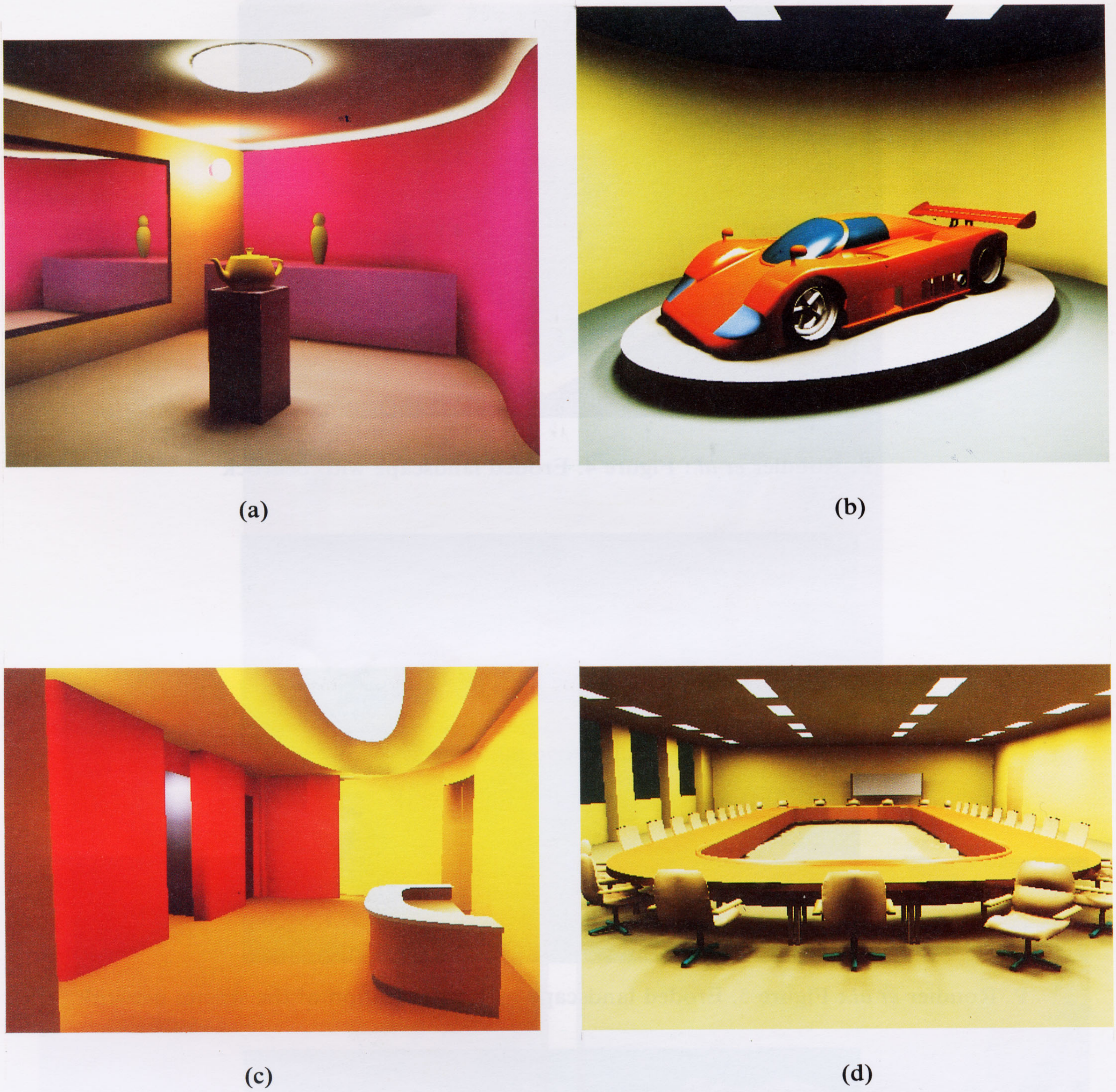
[21] T.Tanaka, T.Takahashi: "Shading with Area Light Sources," Eurographics'91, (1991)pp.235-246.

[22] J.R.Wallace, M.F.Cohen, D.P.Greenberg: "A Two-Pass Solution to The Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods," Computer Graphics, Vol.21, No.4(1987) pp.311-320.

[23] J.R.Wallace, K.A.Elmquist, E.A.Haines: "A Ray Tracing Algorithm for Progressive Radiosity," Computer Graphics, Vol.23, No.3(1989) pp.315-324.

[24] J.E.Warnock: "A Hidden Surface Algorithm for Computer Generated Halftone Pictures," TR 4-15, CS Dept, U. of Utah(1969)

[25] T. Whitted: "An Improved Illumination Model for Shaded Display," CACM, Vol.23, No.6(1980) pp.343-349.



(a)

(b)

(c)

(d)

T. Nishita *et al.*: Figure 8. Application in lighting of an interior taking into account of interreflection.