EUROGRAPHICS '96 / J. Rossignac and F. Sillion (Guest Editors), Blackwell Publishers

Volume 15, (1996), Number 3

Method for Calculation of Sky Light Luminance Aiming at an Interactive Architectural Design

Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tomoyuki Nishita†

Hiroshima University
1-4-1 Kagamiyama,
Higashi-hiroshima, 739 Japan

†Fukuyama University
985 Sanzo, Higashimura-cho,
Fukuyama, 729-02 Japan

Abstract

Recently, computer graphics are frequently used for both architectural design and visual environmental assessment. Using computer graphics, designers can easily compare the effect of the natural light on their architectural designs under various conditions, such as different times of day, seasons, atmospheric conditions (clear or overcast sky) or building wall materials. In traditional methods of calculating the luminance due to sky light, however, all calculation must be performed from scratch if such conditions undergo change. Therefore, to compare the architectural designs under different conditions, a great deal of time has to be spent on generating the images.

This paper proposes a new method of quickly generating images of an outdoor scene, taking into account glossy specular reflection, even if such conditions change. In this method, luminance due to sky light is expressed by a series of basis functions, and basis luminances corresponding to each basis function are precalculated and stored in a compressed form in the preprocess. Once the basis luminances are calculated, the luminance due to sky light can be quickly calculated by the weighted sum of the basis luminances. Several examples of an architectural design demonstrate the usefulness of the proposed method.

Key words and phrases: realistic rendering, sky light, Fourier expansion, vector quantization, glossy specular reflection

1. Introduction

Recently, computer graphics are frequently used for both architectural design and visual environmental assessment. For visual environmental assessment or displaying buildings in outdoor scenes, natural light sources, that is, direct sun light and sky light, are important. Especially, sky light plays an important role to generate photo-realistic images. Using computer graphics, it becomes easy to compare the effect of the natural light on architectural designs under various conditions, such as different times of day, seasons, atmospheric conditions (clear or overcast sky) or building wall materials.

In early computer graphics, only the direct sun light was considered as a natural light source, light from the sky was treated as a constant ambient light emanated from all directions. In 1986, a new method for calculating sky light luminance was proposed [Nishita 86]. Photorealistic images of outdoor scenes can be generated using this method, implying that the sky light plays an important role in the realistic image generation of outdoor scenes. By extending the method further, a method for calculating specular reflection due to sky light was developed [Kaneda 91]. This method pointed out the importance of both the specular and diffuse components in sky lighting in highly realistic rendering.

Calculating the luminance of sky light requires high computational costs because it is equivalent to calculating the luminance of a hemispherical light source. Therefore, in order to reduce computation time, a fast calculation method for determining sky light luminance using a graphics hardware was proposed [Tadamura 93]. Although images can be generated more quickly using this method, sky light luminance must be calculated from scratch every time an image is generated. This means that a great deal of time has to be spent on generating multiple images in order to compare the appearance of a building under various conditions. To address the problem, Nimeroff et al. proposed a fast method of calculating luminance due to sky light, even when the position of the sun changes [Nimeroff 94]. This method, however, requires a great deal of time in the pre-process. Furthermore, the spectral distribution of the sky is not taken into account because the method is limited to a CIE standard distribution of the sky. Consequently, the color variation of buildings can not be reproduced. For example, in the situation where the sun is setting, the color of buildings should turn red. Besides, all luminance calculation must be done from the beginning when materials of walls of buildings are altered.

To address these problems and come up with an interactive architectural design, we have developed a fast calculation method for sky light luminance on a purely diffuse surface when the position of the sun changes [Dobashi 95a]. In the method, the function that expresses the luminance at a calculation point is defined and is approximated by a Fourier series. We call the function a "sky light luminance function." In the pre-process, basis luminances corresponding to each basis function of the Fourier series are calculated and stored. Then the luminance due to sky light can be calculated quickly by the weighted sum of the basis luminances even if the intensity distribution of the sky is altered. However, most architecture consists of surfaces that are not purely diffuse but glossy reflective. Therefore, in order to generate photorealistic images, specular reflection due to sky light must be taken into account.

In this paper, a new method for the fast calculation of sky light luminance, taking into account glossy specular reflection, is proposed. Using the proposed method, images can be quickly generated even when materials of the surfaces, the position of the sun, and atmospheric conditions are altered. Moreover, a vector quantization [Linde 80] introduced for the basis luminances makes it possible to store basis luminances using only a small amount of memory capacity.

In the proposed method, the following conditions are assumed. First, all surfaces are assumed to be purely diffuse or glossy reflective. Second, the interreflection of light between surfaces can be negligible. In outdoor scenes, this assumption is not a serious problem since the interreflection does not have a strong effect.

2. Related Works

Assuming multiple light sources, luminance at a certain point is obtained by calculating the luminance from each light source and summing them. That is, luminance calculation obeys the two following properties [Kajiya 86, Schoenenman 93, Nimeroff 94].

1. Luminance from two light sources is equivalent to the sum of luminances from the individual light sources.
2. Multiplying the luminance from a light source by a factor w is equivalent to multiplying the intensity of the light source by w.

Making use of these properties, we have developed a quick rendering method for interior lighting design under various lighting conditions [Dobashi 95b]. Based on the idea, this paper proposes a fast method for generating images of outdoor scenes when the intensity distribution of the sky and/or surface materials are altered.

In general, the luminance due to sky light, $I_{skylight}$, at a calculation point on a surface can be expressed by the following equation.

$$I_{skylight} = \int_{\Omega} L(\mathbf{s})\kappa(\mathbf{s})d\mathbf{s}, \tag{1}$$

where $\mathbf{s}$ is a unit vector toward an arbitrary direction in the hemisphere above the surface, $L(\mathbf{s})$ is an intensity distribution of the sky, $\kappa(\mathbf{s})$ is a factor determined by the reflectance function of the calculation point and the geometrical relation between the calculation point and a sky element in the direction of $\mathbf{s}$. Ω is the

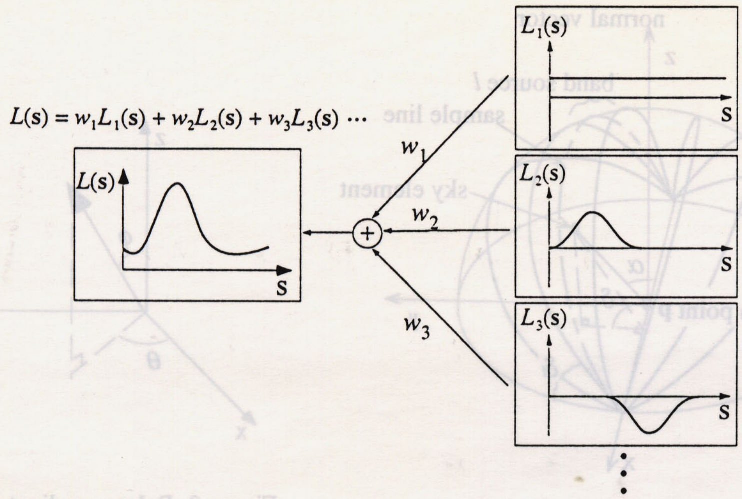


Figure 1: Luminous intensity distribution of the sky and its basis components.

hemispherical integral domain above the surface. Suppose that the intensity distribution of the sky is expressed as the weighted sum of several basic distributions, as shown in Fig. 1. Then the sky light luminance can be calculated as follows. First, in the pre-process, basis luminances due to sky light ,whose distribution is one of the basic distributions expressed as basis functions, are calculated and stored. Next, the luminance due to sky light is calculated by summing the basis luminances with their weights. When the intensity distribution of the sky changes, only the weights need to be recomputed, and the sky light luminance can be obtained quickly by the weighted sum of the basis luminances. This method, however, has its own problems: first, many basic distributions are required to express the intensity distribution of the sky with high accuracy because of a sharp peak around the sun. This results in increasing both the calculation time and memory capacity for the basis luminances.

Nimeroff et al. proposed a fast calculation method for sky light luminance when the position of the sun is altered [Nimeroff 94]. In the method, several components depending on the position of the sun are extracted analytically from the intensity distribution of the sky, and the intensity distribution is expressed as the weighted sum of about 10 basic distributions. However, the method requires a great deal of time for the calculation of basis luminances, since the visibility of the sky is calculated separately, although it can be done as a common computation between basis luminances. Furthermore, CIE standard distribution makes it possible to extract the components depending on the sun position, since CIE standard distribution has an analytical expression. In order to generate photorealistic images, the spectral distribution of the sky must be taken into account. For this purpose, the intensity distribution of the sky should be obtained from the measured data or calculated by a numerical simulation method, such as Klassen's model [Klassen 87], where the intensity of the sky is calculated by integrating the scattered light of the sun due to particles in the atmosphere. In this case, it is difficult to extract the components depending on the sun analytically. Furthermore, when building wall materials change, all calculations must be come out from the beginning.

In the proposed method, the integrated distribution is expanded into a series of basis functions, not the intensity distribution of the sky. This makes it possible to reduce the number of basic distributions. To handle the distributions with no analytical expressions, Fourier cosine series are used for the expansion. In the proposed method, the sky light luminance can be calculated quickly when the position of the sun, the atmospheric conditions, and materials of surfaces change. Cabral et al. proposed an efficient method for environmental mapping on glossy reflective surfaces using a similar idea [Cabral 87]. In their method, however, occlusion effects between objects cannot be handled. In the proposed method, however, the occlusion effects can be handled since basis luminances are calculated taking into account the occlusion. Therefore, the visibility of the sky need only be calculated once to reduce the computation time of the basis luminances.

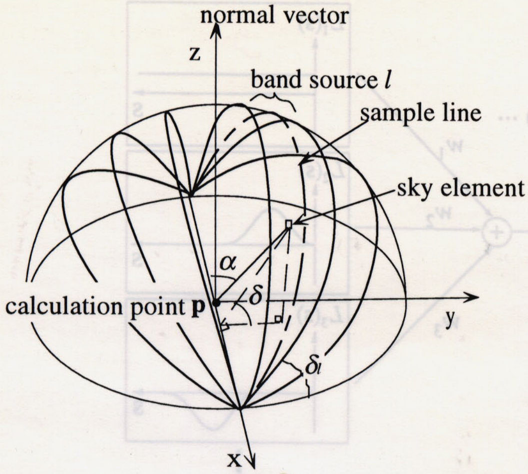


Figure 2: Calculation of sky light luminance.

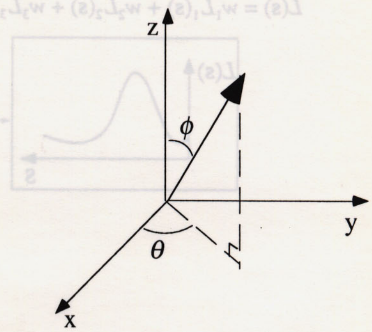


Figure 3: Polar coordinate system.

3. Calculation Method of Sky Light Luminance

3.1 Sky Light Luminance Function and Its Fourier Expansion

Since the *sky dome* is a hemispherical light source of large radius, we can consider the calculation point as always being at the center of this hemisphere. As shown in Fig. 2, let us assume a local coordinate system where its z-axis corresponds to a normal vector of the calculation point, and the direction of a *sky element* to be (α, δ), where $\alpha\,(0 \le \alpha \le \pi)$ is the angle from the x-axis to the sky element and $\delta\,(0 \le \delta \le \pi)$ is the angle from the horizontal plane to the sky element. Then, we define the sky light luminance function as Eq. 2. This luminance function represents the luminance due to a partial region of the sky, $[0, \alpha]$ and $[0, \delta]$ [Nishita 86].

$$F_\lambda(\alpha, \delta; \mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) = \int_0^\alpha \int_0^\delta L_\lambda(\alpha^*, \delta^*; \mathbf{n}, L_0^{(\lambda)})\, \rho_\lambda(\alpha, \delta; \mathbf{v}) \sin^2\alpha^* \sin\delta^*\, d\alpha^* d\delta^*, \tag{2}$$

where λ is the wave length, $\mathbf{n}$ is the normal vector of the calculation point, $L_0^{(\lambda)}$ is the intensity distribution of the sky in the world coordinates, L_λ is the intensity distribution in the local coordinates, $\mathbf{v}$ is a viewing vector from the calculation point, and ρ_λ is a reflectance function of the calculation point. α^* and δ^* are variables for the integral.

In the proposed method, the sky light luminance function is expanded into a Fourier cosine series and approximated with a finite number of cosine functions. That is,

$$F_\lambda(\alpha, \delta; \mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) \approx \sum_{l=1}^{N} \sum_{m=1}^{N} w_{lm}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) \cos(l\alpha) \cos(m\delta), \tag{3}$$

where N is the degree of cosine functions necessary to express the sky light luminance function, and $w_{lm}^{(\lambda)}$ is a weight function corresponding to each cosine function. From the property of the orthogonality of cosine functions, the weight, $w_{lm}^{(\lambda)}$, can be calculated by the following equation.

$$w_{lm}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) = \frac{2}{\pi} \int_0^\pi \int_0^\pi F_\lambda(\alpha, \delta; \mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) \cos(l\alpha) \cos(m\delta)\, d\alpha\, d\delta\,. \tag{4}$$

As shown in Eq. 3, N^2 terms of basis functions are required for approximating the sky light luminance function using cosine functions up to degree N.

3.2 Calculation of Sky Light Luminance Using Sky Light Luminance Function

The sky light luminance, $I^{(\lambda)}_{skylight}(\mathbf{p}, \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0)$, at an arbitrary calculation point, $\mathbf{p}$, can be obtained by integrating luminances due to sky elements visible from the calculation point. Thus, the sky light luminance is given by the following equation.

$$I^{(\lambda)}_{skylight}(\mathbf{p}, \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) = \int_0^{\pi}\int_0^{\pi} H(\alpha, \delta, \mathbf{p}) f_{\lambda}(\alpha, \delta; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0)\, d\alpha d\delta\,, \quad (5)$$

$$f_{\lambda}(\alpha, \delta; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) = \rho_{\lambda}(\alpha, \delta; \mathbf{v})\, L_{\lambda}(\alpha, \delta; \mathbf{n}, L^{(\lambda)}_0) \sin^2\alpha \sin\delta\,,$$

where $H(\alpha, \delta, \mathbf{p})$ is a visibility function that gives 1 if the sky element in the direction of (α, δ) is visible from the calculation point, and otherwise 0. The visible regions of the sky are determined by using the band source method [Nishita 86]. That is, as shown in Fig. 2, the sky dome is divided into several *band sources* to determine the visibility. The center line of the band source is called a *sample line*, and the plane including the sample line and the x-axis is called a *sample plane*. The visible regions of the sky are determined by calculating visible regions of the band sources. The visible regions of a band source are determined by calculating visible sections of its sample line. The visible sections of the sample line are determined by calculating intersections of all objects in the scene with its sample plane [Nishita 86]. Let us assume that the sky dome is divided into n_{band} band sources, and the tilt angle of the sample plane of i-th band source is δ_i ($i = 1, \ldots, n_{band}$). Let visible sections of the sample line of the i-th band source be $(\alpha^{(i)}_j, \alpha^{(i)}_{j+1})$ ($j = 1, \ldots, n^{(i)}_{vis}$), where $n^{(i)}_{vis}$ is the number of visible sections of i-the band source sample line. The sky light luminance at the calculation point is then obtained by the following equation.

$$I^{(\lambda)}_{skylight}(\mathbf{p}, \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) = \sum_{i=1}^{n_{band}} \sum_{j=1}^{n^{(i)}_{vis}} \int_{\delta_i - \Delta_i}^{\delta_i + \Delta_i} \int_{\alpha^{(i)}_j}^{\alpha^{(i)}_{j+1}} f_{\lambda}(\alpha, \delta; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0)\, d\alpha d\delta\,, \quad (6)$$

where $2\Delta_i$ is the angular width of the i-th band source. Substituting equation 6 with equation 2, the following equation is obtained.

$$\begin{aligned} I^{(\lambda)}_{skylight}(\mathbf{p}, \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) = \sum_{i=1}^{n_{band}} \sum_{j=1}^{n^{(i)}_{vis}} \Big(& F_{\lambda}(\alpha^{(i)}_{j+1}, \delta^{+}_i; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) - F_{\lambda}(\alpha^{(i)}_{j}, \delta^{+}_i; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) \\ & - F_{\lambda}(\alpha^{(i)}_{j+1}, \delta^{-}_i; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) + F_{\lambda}(\alpha^{(i)}_{j}, \delta^{-}_i; \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) \Big), \end{aligned} \quad (7)$$

where $\delta^{+}_i = \delta_i + \Delta_i$ and $\delta^{-}_i = \delta_i - \Delta_i$. Equation 7 implies that the integration can be quickly calculated by a method of summed-area table [Crow 84]. From equation 3,

$$I^{(\lambda)}_{skylight}(\mathbf{p}, \mathbf{n}, \mathbf{v}, L^{(\lambda)}_0) \approx \sum_{l=1}^{N} \sum_{m=1}^{N} w^{(\lambda)}_{lm}(\mathbf{n}, \mathbf{v}, L^{(\lambda)}_0)\, B_{lm}(\mathbf{p}), \quad (8)$$

where,

$$\begin{aligned} B_{lm}(\mathbf{p}) = \sum_{i=1}^{n_{band}} \sum_{j=1}^{n^{(i)}_{vis}} \Big(& \cos(l\alpha^{(i)}_{j+1}) \cos(m\delta^{+}_i) - \cos(l\alpha^{(i)}_{j}) \cos(m\delta^{+}_i) \\ & - \cos(l\alpha^{(i)}_{j+1}) \cos(m\delta^{-}_i) + \cos(l\alpha^{(i)}_{j}) \cos(m\delta^{-}_i) \Big). \end{aligned} \quad (9)$$

Equation 8 means that once $B_{lm}(\mathbf{p})$ is precalculated and stored as the basis luminances in the pre-process, the sky light luminance can quickly be obtained by summing $B_{lm}(\mathbf{p})$ with weights. In the proposed method, the basis luminances are calculated for each pixel of an image with a fixed viewpoint. Note that basis luminances does not depend on the wave length, λ. When the intensity distribution of the sky and/or materials of surfaces are altered, only the weights, using Eq. 4, need to be recalculated for each wave length.

3.3 Calculation of Weights

From Eq. 4, the weight, $w^{(\lambda)}_{lm}$, depends on the normal vector and the viewing vector at the calculation point. Generally, these vectors vary at every calculation point, i.e., at every pixel. This implies that the weights must be calculated at every pixel using Eq. 4. In order to calculate the weights, however, first the sky light luminance function has to be evaluated using Eq. 2. Next, using Eq. 4, the weights are calculated.

Evaluating Eqs. 2 and 4 at every pixel, however, takes too much time to generate images for interactive architectural design. To address this problem and reduce the computation time for the weights, a weight table has been generated in the proposed method.

Assuming a polar coordinate system, (θ, ϕ), as shown in Fig. 3, the direction of the normal vector, $\mathbf{n}$, and the viewing vector, $\mathbf{v}$, can be expressed as (θ_n, ϕ_n) and (θ_v, ϕ_v), respectively. In the proposed method, these four variables are sampled at a certain interval, and the weights are calculated at each sampled point and stored as a *weight table*. The weights for arbitrary directions of the vectors are obtained by linear interpolation of the weights stored in the table. However, from the point of memory capacity, it is difficult to store all the weights for all the vector directions. To address this problem, the proposed method makes use of the following property: normal vectors and viewing vectors at each calculation point in an image tend to be biased toward several directions as much architecture are composed of simple geometries such as cubes. This is an indication that it is not always necessary to calculate all the weights for all the vector directions. In the proposed method, only the weights that are necessary to generate the image are calculated and stored. The weights for the same vector directions can be obtained efficiently simply by referring to the table.

4. Compression of Basis Luminances Using Vector Quantization

The number of basis functions is increased in order to approximate the sky light luminance function with high accuracy. This creates a problem, however, as a large amount of memory is required for storing the basis luminances. In this section, a method for compressing the basis luminances using vector quantization [Linde 80] is proposed.

4.1 Basis Luminance Vector

In the proposed method, N^2 basis luminances, $B_{lm}(\mathbf{p}_i)$, are stored for each pixel where $\mathbf{p}_i$ is a three-dimensional coordinate corresponding to i-th pixel in an image. These N^2 basis luminances are treated as an N^2 dimensional vector. We call the N^2 dimensional vector a *basis luminance vector*. The vector quantization is applied to the basis luminance vector. The basis luminance vector, $\mathbf{b}_i$, for i-th pixel is defined as follows.

$$\mathbf{b}_i = \left\{ B_{00}(\mathbf{p}_i), B_{01}(\mathbf{p}_i), \cdots, B_{NN}(\mathbf{p}_i) \right\}. \tag{10}$$

4.2 Vector Quantization

Basis luminance vectors of neighboring points on the same plane are considered to be analogous to each other. This means that basis luminance vectors for all pixels can be classified into several classes, as shown in Fig. 4. Note that, in Fig. 4, basis luminance vectors are depicted as three-dimensional vectors. In vector quantization, the vector that represents each class is selected for each class. The vector is called a *reproduction vector*. All the basis luminance vectors in the same class are approximated by the reproduction vector of the class. A set of reproduction vectors is called a *codebook* [Linde 80].

An algorithm called the LBG algorithm [Linde 80] has been proposed for making a good codebook. The LBG algorithm for basis luminance vectors is as follows.

1. Initial reproduction vectors are generated in N^2 dimensional space.
2. The distance, d, between basis luminance vectors and reproduction vectors is calculated and basis luminance vectors are classified into one of the classes with the minimum distance. This process is repeated for all the basis luminance vectors in an image.
3. Reproduction vectors are replaced by average basis luminance vectors of each class.
4. If there are such classes into which no basis luminance vectors are classified, the classes are deleted. If all the classes have basis luminance vectors, stop the procedure. Otherwise go to step 5.
5. New reproduction vectors are generated as the same number of deleted classes in step 4. Go to step 2.

The distance between two arbitrary vectors, $\mathbf{b}_i$ and $\mathbf{b}_j$, is defined as follows [Linde 80].

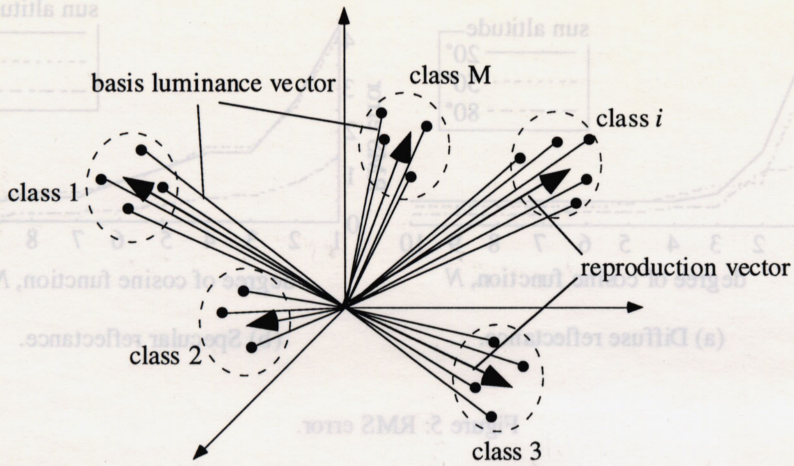


Figure 4: Vector quantization.

$$d(\mathbf{b}_i, \mathbf{b}_j) = \sum_{l=1}^{N} \sum_{m=1}^{N} \left| B_{lm}(\mathbf{p}_i) - B_{lm}(\mathbf{p}_j) \right|^2. \tag{11}$$

Using the LBG algorithm, the codebook is generated so that the square error is minimum [Linde 80]. In order to apply the LBG algorithm to basis luminance vectors, methods for generating initial reproduction vectors (step 1) and new reproduction vectors (step 5) are proposed as follows.

The arrangement of initial reproduction vectors has to reflect the statistical character of input vectors to rapidly converge the LBG algorithm. And, to make a good codebook, initial reproduction vectors should be arranged uniformly in N^2 dimensional space. In the proposed method, initial reproduction vectors are determined by randomly selecting basis luminance vectors in an image. This makes it possible to reflect the statistical character of input vectors on initial reproduction vectors. Furthermore, a vector whose direction is opposite that of the randomly selected vector is also used as a initial reproduction vector. This prevents the initial reproduction vectors from being biased toward certain directions.

The variance of a class is taken into account in generating new reproduction vectors. If the variance of distance between a reproduction vector and the basis luminance vectors classified into the class has a large value, the class should be divided into smaller sub-classes in order to suppress the approximation error. In the proposed method, the larger the variance of a class, the smaller sub-classes the class is divided into. That is, each class is divided into sub-classes in proportion to its variance. New reproduction vectors, $\mathbf{b}'_k$, are generated around the original reproduction vector, $\mathbf{b}_i$, using the following equation.

$$\mathbf{b}'_k = \mathbf{b}_i + \mathbf{e}_k, \quad (k = 1, \cdots, n_{div,i}), \tag{12}$$

where $\mathbf{e}_k$ is an infinitesimal vector generated randomly, and $n_{div,i}$ is the number of division of class i. By repeating steps 2 through 5 in the LBG algorithm using the new reproduction vectors, classes with large variances are divided into smaller sub-classes.

4.3 Weighted Sum of Basis Luminances

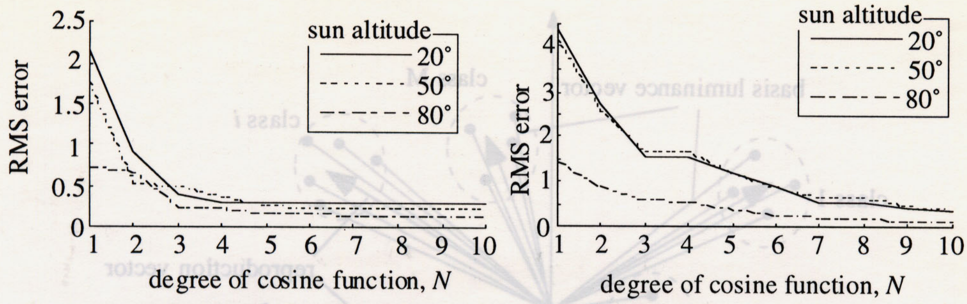
In this section, a method for calculating the sky light luminance using vector-quantized basis luminances is described. Let us define a weight vector as the following equation.

$$\mathbf{w}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) = \left\{ w_{00}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}), w_{01}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}), \cdots, w_{NN}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) \right\}, \tag{13}$$

Then, from Eq. 8, the sky light luminance at calculation point, $\mathbf{p}_i$, can be obtained by calculating the dot product between the basis luminance vector, $\mathbf{b}_i$, and the weight vector. That is,

$$I_{skylight}^{(\lambda)}(\mathbf{p}_i, \mathbf{n}, L_0^{(\lambda)}) = \mathbf{w}^{(\lambda)}(\mathbf{n}, \mathbf{v}, L_0^{(\lambda)}) \bullet \mathbf{b}_i \tag{14}$$

where • means the dot product between two arbitrary vectors.



(a) Diffuse reflectance. (b) Specular reflectance.

Figure 5: RMS error.

5. Image Generation

The proposed algorithm for generating images is as follows.

0. In the pre-process, the basis luminance, $B_{lm}(\mathbf{p}_i)$, is calculated for each pixel and stored after the vector quantization.
1. For each pixel, repeat the following steps.
   (a) The sky light luminance function is calculated using Eq. 2. Then the function is expanded into a Fourier cosine series. The weights for each basis function are calculated using the method described in section 3.3.
   (b) The sky light luminance is obtained by calculating the inner product between the basis luminance vector and the weight vector.
   (c) Luminance due to the direct sun light is calculated. The color of the pixel is determined by summing the direct sun light luminance and the sky light luminance.

Once the basis luminances are calculated in the pre-process, images can be quickly generated by repeating step 1, even if conditions such as the position of the sun, the conditions of the sky, and materials of surfaces are altered.

6. Discussion on Approximation of Sky Light Luminance Function

In this section, the accuracy of approximating the sky light luminance function using the Fourier cosine series is discussed.

To investigate the approximation error, RMS error is calculated by changing the cosine function degree (N in Eq. 3). The intensity distribution of the sky is assumed to be the CIE standard clear sky [CIE 73], and the sky light luminance function at a point on a horizontal plane is calculated using Eq. 2. Then the function is approximated by the Fourier cosine series. Figure 5 shows RMS error between the original function and the approximated function. Figure 5(a) shows the error when the altitude of the sun is 20, 50, and 80 degrees and the surface reflectance is assumed to be purely diffuse. In Fig. 5(b), the reflectance of the surface is assumed to be glossy specular (aluminum). The specular reflectance is calculated using the Cook-Torrance model [Cook 82]. The wave length, λ, is sampled at 675 [nm].

As shown in Fig. 5, the error decreases as the number of basis functions increases. In the case of diffuse reflection, the error is converged early since the sky light luminance function has a lenient distribution in this case. On the other hand, as shown in Fig. 5(b), a larger number of basis functions is required in order to approximate the sky light luminance function for glossy specular reflection. This is because the high frequency components are contained within the specular reflection. As shown in Fig. 5, the error is less than

Table 1: Computation time [min.].

	Previous Method	Proposed Method
Basis luminance	-	315.7
Fig. 6	101.1	3.0
Fig. 7(a)	103.4	3.4
Fig. 7(b)	96.6	3.5
Fig. 7(c)	103.7	3.6
Fig. 7(d)	75.1	3.5
Fig. 7(e)	97.0	3.4
Fig. 7(f)	74.6	3.6

machine : SiliconGraphics PowerIndigo^2.

0.5 when the degree of cosine function is greater than 3 (diffuse reflection) or 10 (specular reflection). These figures show that the sky light luminance function can be approximated with a high degree of accuracy using the Fourier cosine series.

7. Examples

To investigate the usefulness of the proposed method, it is applied to architectural designs. In the following examples, the wave length is sampled at 675, 520 and 460 [nm], figures that correspond to RGB components. First, an image of a building is generated using both the proposed method and the previous method [Kaneda 91]. In Fig. 6,* only the sky light luminance is calculated in order to provide an appropriate comparison. Figure 6(a) shows the image formulated by the previous method. The altitude of the sun is set to 50 degrees and the reflectance of the walls is assumed to be purely diffuse. The intensity distribution of the sky is calculated using Klassen's method [Klassen 87]. Figure 6(b) is generated using the proposed method. The degree of cosine functions, N, is 10, i.e., the number of basis functions is 100. The size of the image is 512X420. The memory capacity required for the basis luminance is 86 MB without the vector quantization. With the vector quantization, the required capacity is decreased to 5 MB. The number of reproduction vectors is 10000. The maximum relative error between Figs. 6(a) and 6(b) is 7 percent. This means that the proposed method can generate images with almost the same quality as the previous method.

Figure 7* shows images of the same building under different conditions. In Figs. (a) through (c), the walls of the building are made of tiles, and the wall material used in Figs. (d) through (f) is aluminum. Figures (a) and (d) are the case of morning (the sun altitude is 20 degrees), figures (b) and (e) is in a daytime (50 degrees), and figures (c) and (f) are at sunset (10 degrees). The reflectance of aluminum is more specular than that of tiles. Therefore, in Figs. (d) through (f), the sky is well-reflected in the walls of the building. Especially, in Fig. (f), the color of the upper part of the building is dark blue while the color of the lower part is orange. In Fig. (c), such color variations are less visible. In these figures, cloud rendered using the method described in [Kaneda 91] does not have any effect on the luminance calculation of the building. As shown in these figures, the color of the building changes as the sun altitude changes, and highlights due to specular reflection also change depending on the sun position and the materials.

The comparison of the computation time for generating the images is shown in Table 1. All the images are generated on a SiliconGraphics PowerIndigo^2. As shown in Table 1, once the basis luminances are calculated, the proposed method can generate an image in about 3 minutes even when the position of the sun and materials of surfaces change. In contrast, the previous method requires about 100 minutes for generating each image. Moreover, the calculation time of the basis luminances is only about 3 times longer than the time needed to generate an image using the previous method.

*** See pages C–465 and C–466 for figures 6 and 7.**

8. Conclusions

By expanding the sky light luminance function into a Fourier cosine series, we have proposed a method for quickly calculating the sky light luminance when the intensity distribution of the sky and/or surface materials are changed. In the proposed method, the basis luminances are calculated and stored in the preprocess and the sky light luminance can be obtained quickly by the weighted sum of the basis luminances. Applying the vector quantization to the basis luminances, the memory capacity required for the basis luminances can be greatly reduced. Using the proposed method, images can be generated quickly even when many conditions such as time in a day, seasons, atmospheric conditions (clear or overcast sky), and building wall materials are altered. This makes it easy to investigate the effects of such conditions on architectural designs.

As future work, a further speeding up of the process is needed. Although the proposed method can generate images 25 times faster than the previous method [Kaneda 91], this is not fast enough for the systems that require real-time rendering, such as virtual reality systems. Besides, it is important to develop suitable graphical user interfaces for an interactive system of visual environmental assessment.

References

[Cabral 87] B. Cabral, N. Max and B. Springmeyer, *Bidirectional Reflection Functions from Surface Bump Maps*, Computer Graphics, Vol. **21**, No. 4, pp. 273-281 (1987).

[Cook 82] R. L. Cook and K. E. Torrance, *A Reflectance Model for Computer Graphics, ACM Trans. on Graphics*, Vol. **19**, No. 3, pp. 7-24 (1982).

[Crow 84] Franklin C. Crow, *Summed-Area Tables for Texture Mapping*, Computer Graphics, Vol. 18, No. 3, pp. 207-212 (1984).

[CIE 73] CIE Technical Comittee 4.2: *Standardization of Luminance Distribution on Clear Skies*, CIE Publication, No. 22, Comission International de l'Eclairaze, Paris pp. 7 (1973).

[Dobashi 95a] Y. Dobashi, K. Kaneda, H. Yamashita, and T. Nishita, *A Quick Rendering Method for Outdoor Scenes Using Sky Light Luminance Function Expressed with Basis Functions*, The Journal of The Institute of Image Electronics Engineers of Japan, Vol. **24**, No. 3, pp. 196-205 (1995) (in Japanese).

[Dobashi 95b] Y. Dobashi, K. Kaneda, H. Nakatani, H. Yamashita, and T. Nishita, *A Quick Rendering Method Using Basis Functions for Interactive Lighting Design*, Computer Graphics Forum, Vol. **14**, No. 3, pp. 229-240 (1995).

[Kajiya 86] J. T. Kajiya, *The Rendering Equation*, Computer Graphics, Vol. **20**, No. 4, pp. 143-150 (1986).

[Kaneda 91] K. Kaneda, T. Okamoto, E. Nakamae and T. Nishita, *Photorealistic Image Synthesis for Outdoor Scenery under Various Atmospheric Conditions*, The Visual Computer, Vol. **7**, No. 5-6, pp. 247-258 (1991).

[Klassen 87] R. V. Klassen, *Modeling the Effect of the Atmosphere on Light*, ACM Trans. on Graphics, Vol. **6**, No. 3, pp. 215-237 (1987).

[Linde 80] Y. Linde, A. Buzo, and R M. Gray, *An Algorithm for Vector Quantizer Design*, IEEE Trans. on Communications, Vol. **28**, No. 1, pp. 84-95 (1980).

[Nimeroff 94] J. S. Nimeroff, E. Simoncelli, and J. Dorsey, *Efficient Re-rendering of Naturally Illuminated Environments*, Proceedings of 5th Eurographics Workshop on Rendering, pp. 359-373 (1994).

[Nishita 86] T. Nishita and E. Nakamae, *Continuous Tone Representation of Three-Dimensional Objects Illuminated by Sky Light*, Computer Graphics, Vol. **20**, No. 4, pp. 125-132 (1986).

[Schoenenman 93] C. Schoenenman, J. Dorsey, B. Smits, J. Arvo, and D. P. Greenberg, *Painting with Light*, Computer Graphics (Proceedings of SIGGRAPH'93), pp. 143-146 (1993).

[Tadamura 93] K. Tadamura, E. Nakamae, K. Kaneda, H. Yamashita, and T. Nishita, *Modeling of Skylight and Rendering of Outdoor Scenes*, Computer Graphics Forum, Vol. **12**, No. 3, pp. 189-200 (1993).