PAPER

Clustering Environment Lights for an Efficient All-Frequency Relighting

Henry JOHAN†∗a), *Nonmember and* **Tomoyuki NISHITA**†b), *Member*

SUMMARY We present a novel precomputed radiance transfer method for efficient relighting under all-frequency environment illumination. Environment illumination is represented as a set of environment lights. Each environment light comprises a direction and an intensity. In a preprocessing step, the environment lights are clustered into several clusters, taking into account only the light directions. By experiment, we confirmed that the environment lights can be clustered into a much smaller number of clusters than their original number. Given any environment illumination, sampled as an environment map, an efficient relighting is then achieved by computing the radiance using the precomputed clusters. The proposed method enables relighting under very high-resolution environment illumination. In addition, unlike previous approaches, the proposed method can efficiently perform relighting when some regions of the given environment illumination change.

key words: *precomputed radiance transfer, all-frequency relighting, clustering environment lights*

1. Introduction

Producing photorealistic images is one of the most important goals in computer graphics. Recently, rendering photorealisitic images under environment illumination has gained much attention. However, in the case of environment illumination, we consider lights incident from all directions, resulting in a high rendering cost when using conventional methods, such as ray tracing. This becomes a serious problem in a dynamic lighting environment such as lighting design, where the lighting environment is continuously manipulated and rendering has to be performed for each change.

With the emergence of the precomputed radiance transfer methods, fast relighting of a rigid scene under environment illumination is realized. The idea is to represent the light transfer phenomena on each vertex of the objects as a light transfer function. The light transfer function and the lighting environment are approximated using several basis functions, such as spherical harmonics [16] or wavelets [11], [12]. The radiance at each vertex is then computed using these basis functions.

The limitation in using the spherical harmonics is that they can only be applied to a low-frequency lighting environment, resulting in low-frequency shadows. Using wavelets overcomes this problem, realizing all-frequency shadows in the rendered images. In this paper, all-frequency is used with a meaning of frequency-independent, that is, from low-frequency up to high-frequency. Therefore, all-frequency shadows include shadows with blurred boundaries (low-frequency shadows) and sharp boundaries (high-frequency shadows). However, due to the computational costs and storage requirements, the wavelet approach can only deal with environment illumination, represented as cubical environment map, up to a resolution of $6 \times 256 \times 256$. Another limitation is that full recomputation of relighting is required, even when only portions of the environment illumination change.

In this paper, we present a method for fast relighting that overcomes the two limitations mentioned above. Our method can deal with an all-frequency lighting environment and thus we can produce all-frequency effects in the rendered images, such as soft and hard shadows. If the scene consists of only diffuse objects, the viewpoint can be changed during the relighting. For a scene consisting of objects with arbitrary bidirectional reflectance distribution functions (BRDFs), the viewpoint is fixed.

Unlike all the previous approaches, instead of each vertex, we define a light transfer vector for each environment light. The elements of the light transfer vector are the light transfer functions at the vertices of the objects. We have observed that environment lights in proximity, whose directions are similar to each other, tend to have similar light transfer vectors. Using this fact, therefore, we cluster the environment lights based on the similarities in their light transfer vectors. We will show that the environment lights can be clustered into a small number of clusters compared to the original number of environment lights, whilst preserving the all-frequency effects in the rendered images. Given an environment map as the environment illumination, using the precomputed clusters, an efficient relighting can be achieved.

Our approach by clustering the environment lights also has the following advantages.

- Increasing the resolution of the environment illumination is only increasing the number of environment lights whose directions are similar, that is their light transfer vectors also tend to be simi-

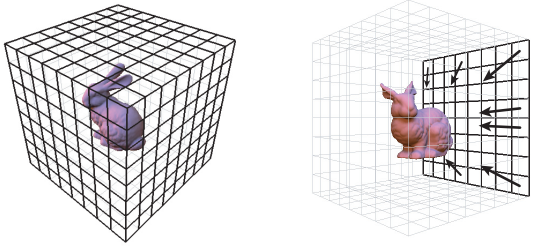
---

†The authors are with the University of Tokyo.

∗Presently, the author is with the Nanyang Technological University.

a) E-mail: henry@is.s.u-tokyo.ac.jp

b) E-mail: nis@is.s.u-tokyo.ac.jp



(a) (b)

Fig. 1 (a) Relighting a scene under environment illumination sampled as lights emanating from a cube. (b) Directions (black arrows) of several environment lights.

lar. Therefore, increasing the resolution does not greatly influence the clustering result and results in only a relatively small increase in the number of clusters. As a result, an efficient relighting under very high-resolution environment illumination is possible.

- When the light intensities in regions of the given environment map changed, the radiances at all vertices are updated by recomputing only the radiance due to the clusters that are affected by the changes. That is, full recomputation for the relighting is not required, resulting in an efficient relighting.

2. Related Work

Dobashi *et al.* [4] presented a method for fast rendering under skylight illumination using basis functions. A fast relighting method for a human face by capturing its reflectance field using a device called light stage was proposed by Debevec *et al.* [3]. Ramamoorthi and Hanrahan [13], [14] proposed methods to rendered scenes under environment illumination in real time using spherical harmonics. However, their methods did not take shadows into consideration. Sloan *et al.* [16] proposed a precomputed radiance transfer method using spherical harmonics as the basis functions for rendering various effects such as soft shadows, direct and indirect illumination and caustics. Kautz *et al.* [6] proposed a method for rendering scenes with arbitrary BRDFs. Lehtinen and Kautz [9] presented a method for rendering glossy objects efficiently. Sloan *et al.* used clustered principle component analysis for compressing the precomputed data [15] and also presented a method to handle meso-structures on surfaces [17]. The limitation of these approaches is that they can only deal with low-frequency lighting environment.

Ng *et al.* [11], [12] used wavelets and achieved relighting under all-frequency environment illumination. By using a separable BRDF approximation [5], Wang *et al.* [19], [20] and Liu *et al.* [10] extended the work in [11] to handle objects with a complex BRDF. These approaches, however, can only handle environment illumination up to a certain moderate resolution. The highest resolution of 6 × 256 × 256 is achieved using the method in [12]. The important limitation is that, even when only some regions of the environment illumination are changed, these approaches have to perform full recomputation for the relighting.

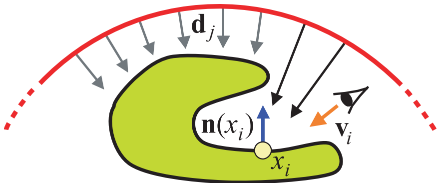


Fig. 2 Radiance computation at vertex x_i.

Methods to accelerate rendering by reducing the environment illumination into hundreds of directional light sources were proposed by Agarwal *et al.* [1], Kollig and Keller [7]. However, these methods still take a few minutes to render an image. Moreover, given an environment map, these methods must convert the environment map into a set of directional lights before the rendering can be performed.

Recently, Walter *et al.* [18] and Kristensen *et al.* [8] presented methods to accelerate the rendering under many lights by clustering them. These approaches are similar to our approach in the manner of reducing the computational cost by clustering the lights. We will discuss the differences between our method and these methods later. Our method clusters environment lights based only on their directions, and thus the clustering results can be applied to any environment illumination. In addition, our method can relight a scene in several seconds.

3. Basic Idea

We assume that the environment illumination is sampled as environment lights coming from a six-sided cube (see Figure 1(a)) with resolution of $6 \times R \times R$, where R is an arbitrary positive integer. In other words, each face of the cube is subdivided into $R \times R$ regions. The directions of the environment lights are determined to be those from the center of regions on the faces of the cube to the center of the cube (see Figure 1(b)).

3.1 Radiance due to environment illumination

Assume that there are N vertices in the scene. The radiance B at the vertex x_i $(i = 1, ..., N)$, lit by direct illumination from environment illumination, is computed as follows (see Figure 2).

$$B(x_i,\mathbf{v}_i)=\sum_{j=1}^{M} L(\mathbf{d}_j)\,V(x_i,\mathbf{d}_j)\,\rho(x_i,\mathbf{d}_j,\mathbf{v}_i)\,\max(\mathbf{d}_j\cdot\mathbf{n}(x_i),0). \tag{1}$$

$\mathbf{v}_i$ is the viewing direction, $M = 6R^2$ is the number of environment lights, $\mathbf{d}_j$ $(j = 1, ..., M)$ are the incident light directions, $L(\mathbf{d}_j)$ is the incident light intensity from the direction $\mathbf{d}_j$, $V(x_i, \mathbf{d}_j)$ is the binary visibility function, indicating if the light from the direction $\mathbf{d}_j$ reachs x_i, $\mathbf{n}(x_i)$ and $\rho(x_i, \mathbf{d}_j, \mathbf{v}_i)$ are the normal vector and the BRDF at x_i, respectively.

In a similar manner to Ng *et al.*[11], we consider two cases.

- **Geometry relighting:** If all the objects in the scene are diffuse surfaces, ρ only depends on surface location and we define a transfer function as

$$T_j(x_i) = V(x_i, \mathbf{d}_j)\, \rho(x_i)\, \max(\mathbf{d}_j \cdot \mathbf{n}(x_i), 0). \quad (2)$$

Since we do not fix the viewing direction, in the relighting process, we can interactively change the viewpoint.

- **Image relighting:** When the scene consists of objects with arbitrary BRDFs, we fix the viewpoint and define the transfer function as

$$T_j(x_i) = V(x_i, \mathbf{d}_j)\, \rho(x_i, \mathbf{d}_j, \mathbf{v}_i)\, \max(\mathbf{d}_j \cdot \mathbf{n}(x_i), 0). \quad (3)$$

In both cases, the radiance computations are not dependent on the viewing direction. Using $T_j(x_i)$, Equation (1) can be rewritten as

$$B(x_i) = \sum_{j=1}^{M} T_j(x_i)\, L(\mathbf{d}_j). \quad (4)$$

Then, the radiance of all the vertices can be written in matrix notation as follows.

$$\mathbf{B} = \mathbf{T}\,\mathbf{L},$$

$$\begin{pmatrix} B(x_1) \\ \vdots \\ B(x_N) \end{pmatrix} = \begin{pmatrix} T_1(x_1) & \dots & T_M(x_1) \\ \vdots & \ddots & \vdots \\ T_1(x_N) & \dots & T_M(x_N) \end{pmatrix} \begin{pmatrix} L(\mathbf{d}_1) \\ \vdots \\ L(\mathbf{d}_M) \end{pmatrix}, \quad (5)$$

where $\mathbf{B}$ is a N-dimensional vector of radiance of all the vertices, $\mathbf{T}$ is a $N \times M$-transfer matrix, and $\mathbf{L}$ is a M-dimensional vector of the light intensities.

3.2 Efficient radiance computation

When the resolution R of the cube is large, it is computationally expensive to evalute Equation (4) for each vertex. Ng *et al.*[11] perform the Haar wavelet transform to each row of matrix $\mathbf{T}$ and illumination vector $\mathbf{L}$ in Equation (5). By employing non-linear approximation, efficient radiance computation at each vertex is achieved. Their approach, however, has the following limitations.

1. Increasing the resolution R results in the increases of both the computational cost and the storage requirements of the precomputed data, which are proportional to the increase in R. This limits their approach for handling very high-resolution environment illumination.
2. When the light intensities in the given environment map are locally changed or editted, the full computation of the relighting must be performed as if for a new environment map.

We propose an efficient method to compute the radiances at the vertices. In addition, our method overcomes the above-mentioned limitations. Instead of the row of $\mathbf{T}$, we pay attention to the column of $\mathbf{T}$. The j-th column of $\mathbf{T}$ represents the transfer function of each vertex with respect to the j-th environment light. In this paper, we will define the j-th column of $\mathbf{T}$ as *light transfer vector* $\mathbf{T}_j$ which is a N-dimensional vector.

$$\mathbf{T}_j = \begin{pmatrix} T_j(x_1) \\ \vdots \\ T_j(x_N) \end{pmatrix}. \quad (6)$$

Using the light transfer vectors, the radiances at all vertices $\mathbf{B}$ can be expressed as

$$\mathbf{B} = \sum_{j=1}^{M} L(\mathbf{d}_j)\, \mathbf{T}_j. \quad (7)$$

The cost of computing Equation (7) can be reduced as follows. We pay attention to the fact that when the incident directions of several environment lights are similar, then there is a high probability that their light transfer vectors are also similar. Based on this fact, we cluster the environment lights, based on the similarities of their light transfer vectors, into m clusters C_k $(k = 1, ..., m)$, where $m < M$. Then, the radiances at all vertices can be approximated as

$$\mathbf{B} \approx \mathbf{B}_A = \sum_{k=1}^{m} L_{C_k}\, \mathbf{T}_{C_k}. \quad (8)$$

Each cluster C_k consists of environment lights whose light transfer vectors are similar. L_{C_k} is the radiant intensity of cluster C_k which is the sum of the light intensities of the environment lights in C_k. $\mathbf{T}_{C_k}$ is the light transfer vector of cluster C_k which is computed based on the light transfer vectors of the environment lights in C_k. By performing the clustering based on the similarities of the light transfer vectors, the approximated radiance $\mathbf{B}_A$ will be close enough to the real radiance $\mathbf{B}$, resulting in all-frequency effects being realized in the rendered images. Furthermore, note that Equation 8 has a lower computational cost than Equation 7.

As shown in Equations (2) and (3), transfer functions depend on light directions but not on light intensities. Therefore, after the environment lights are clustered, we can apply the clustering results to any given environment maps in the relighting process.

3.3 Overview of the proposed method

The proposed method comprises two steps.

1. Preprocessing: Given R, the environment lights in the environment illumination with a resolution of $6 \times R \times R$ are clustered based on the similarities of their light transfer vectors. For each cluster C_k, its light transfer vector $\mathbf{T}_{C_k}$ is also computed.
2. Relighting: Given an environment map, the scene is relighted. We also propose an efficient method to relight a scene when the light intensities in the given environment map are locally changed.

4. Clustering Environment Lights

The simplest approach to clustering the environment lights is to compute the light transfer vectors of all the lights and then to perform the clustering using all the light transfer vectors. However, this approach is not efficient, particularly when the resolution of the environment illumination is high, because the computational cost of computing all the light transfer vectors is high and the clustering of many high-dimensional vectors is not easy.

However, since the environment lights whose incident directions are close to each other tend to have similar transfer vectors, we only choose a few samples from all the lights in our approach, and then perform the clustering based on these samples.

The clustering is performed for each cube face of the environment illumination independently as follows. We initialize the clustering domain with the cube face of the environment illumination. The environment lights in the clustering domain are clustered using the following procedures.

1. Select several environment lights as samples from the clustering domain (Section 4.1).
2. Compute the light transfer vectors for the samples (Section 4.2).
3. If these light transfer vectors are similar to each other, then all the lights in the clustering domain are treated as one cluster. Otherwise, quadrisect the current clustering domain into four smaller domains and process each new smaller domain independently (Section 4.3).

4.1 Selecting environment light samples

Our strategy is to choose a set S of $s \times s$ number of samples uniformly inside the clustering domain. Figure 3(a) shows the case where 3×3 (nine) samples have been chosen. If the domain has less than s^2 samples, then we use all the environment lights inside the domain as samples.

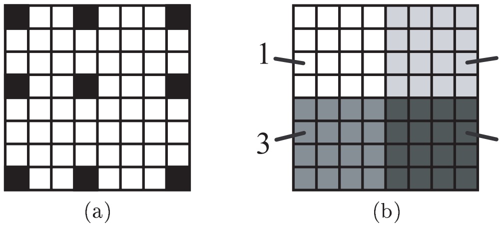


Fig. 3 (a) Selecting samples of environment lights from the clustering domain. (b) Quadrisecting the region when the approximation error is large.

4.2 Computing light transfer vectors

For each environment light sample in set S, its light transfer vector is computed by computing the transfer function at each vertex using Equation (2) or (3). In the case of image relighting (see Section 3.1), for each light sample, three light transfer vectors are computed for the red, green, and blue components.

During the clustering, several environment lights might be chosen as samples on several occasions. In this case, it is inefficient to compute their light transfer vectors multiple times. Therefore, the computed light transfer vector for a specific environment light is stored and is reused whenever required. Only when the environment light is no longer to be chosen as sample, that is the environment light has been classified to a specific cluster, its light transfer vector is discarded.

4.3 Clustering test

We compute the light transfer vector $\mathbf{T}_C$ for the clustering domain such that the sum of differences between $\mathbf{T}_C$ and the light transfer vectors of the environment lights at the samples, $\mathbf{T}_l$, $l \in S$ is minimized.

$$\min \sum_{l \in S} \|\mathbf{T}_C - \mathbf{T}_l\|^2. \tag{9}$$

The light transfer vector $\mathbf{T}_C$ that satisfies Equation (9) can be obtained as follows.

$$\mathbf{T}_C = \frac{\sum_{l \in S} \mathbf{T}_l}{|S|}, \tag{10}$$

where $|S|$ is the cardinality of set S which is the number of samples in S.

To decide whether we can treat all the environment lights in the current clustering domain as one cluster, we perform the following test on all samples $l \in S$.

$$\frac{\|\mathbf{T}_C - \mathbf{T}_l\|}{N} < \epsilon. \tag{11}$$

ϵ is a given threshold. If all the samples satisfy the above condition, then we treat all the environment lights in the current clustering domain as one cluster, with $\mathbf{T}_C$ as the approximated light transfer vector. Otherwise, we quadrisect the current clustering domain (see Figure 3(b)) and process each new domain independently.

5. Relighting

Assume that the environment lights comprising an environment illumination with resolution $6 \times R \times R$ are clustered into m clusters C_k $(k = 1, ..., m)$ for a particular scene.

5.1 Computing the radiance

Given an environment map, we first compute the intensities of the environment lights at the cube of the environment illumination from the given map. Then, based on the clustering information, we compute the radiant intensity L_{C_k} of each cluster by summing the intensities of lights that belong to cluster C_k. Finally, the radiances at all the vertices are determined by computing Equation (8).

5.2 Local changes in the environment map

Let $\mathbf{B}_A$ be the radiances at all vertices illuminated under a specific environment map. Assume that some portions of the map are changed, meaning that the radiant intensities of some clusters changed. Let C_t $(1 \leq t \leq m)$ be one of such clusters and its radiant intensity has changed from L_{C_t} to L'_{C_t}. The new radiances at all vertices $\mathbf{B}'_A$ due to the change in the radiant intensity of cluster C_t can be efficiently computed as follows.

The radiances at the vertices before the change are

$$\mathbf{B}_A = L_{C_t}\,\mathbf{T}_{C_t} + \sum_{k \neq t} L_{C_k}\,\mathbf{T}_{C_k}. \tag{12}$$

The radiance due to clusters other than C_t are as follows.

$$\sum_{k \neq t} L_{C_k}\,\mathbf{T}_{C_k} = \mathbf{B}_A - L_{C_t}\,\mathbf{T}_{C_t}. \tag{13}$$

The radiances at the vertices after the change can be determined as

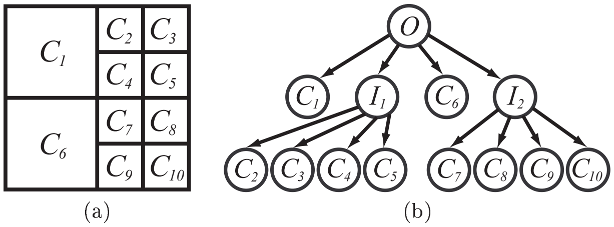
$$\mathbf{B}'_A = L'_{C_t}\,\mathbf{T}_{C_t} + \sum_{k \neq t} L_{C_k}\,\mathbf{T}_{C_k}. \tag{14}$$

By substituting Equation (13) into Equation (14), the new radiances can be computed efficiently as

$$\mathbf{B}'_A = \mathbf{B}_A + (L'_{C_t} - L_{C_t})\,\mathbf{T}_{C_t}. \tag{15}$$

That is, we do not have to recompute the contributions of all the clusters and sum up the results. Instead, we just compute those changes due to cluster C_t (the second term at the right hand side of Equation (15)) and use the results to update the radiance.

The only problem remaining is how to find the clusters whose radiant intensities changed when some of the regions of the given environment map have changed. In the clustering process, the clustering domain is recursively quadrisected. As a result, a quadtree can be used to represent the clusters distribution (see Figure 4). The leaf nodes of the quadtree represent the clusters. Assume that D is the bounding box of a region in the environment map that changed. Clusters whose radiant intensities changed can be easily determined by traversing the quadtree and checking if D intersects the region represented by a node. If D intersects the region represented by a leaf node, the radiance is updated according to Equation (15) for the cluster corresponding to this leaf node.



(a) (b)

Fig. 4 (a) An example of the clustering result at one face of the cube and (b) its cluster distribution represented using a quadtree. O is the root, I represent the internal nodes, and C represent the leaf nodes which are clusters.

Table 1 Statistics of the three scenes used in the experiments.

	Bunny	Dragon	Buddha
Vertices	73196	137499	69577
Resolution R	1024	1024	256
Clusters	9831	5337	4701
Precomputation data (MB)	405	326	623
Precomputation time (min)	50	35	8
Relighting time (sec)	6.5	8.0	3.5

6. Results and Discussions

We performed experiments using three scenes, a Stanford Bunny scene, a Dragon scene, and a Happy Buddha scene. The Bunny and Dragon scenes consisted of white diffuse objects while the Happy Buddha scene consisted of brown glossy objects. The statistics of these scenes are shown in Table 1. For compression, we quantized each element of the light transfer vector into one byte and discarded all the zero elements. We used the environment map images provided by Paul Debevec [2] in our experiments. The computations were performed on a computer with a 3.4 GHz Pentium 4. The experiment results show that our method can relight at interactive rates. For comparison, the rendering time for Bunny scene by exact integration of all environment lights is about five hours for environment illumination with resolution 256.

We visualized the light transfer vector by setting the light transfer function at each vertex as its radiance and then rendering the scene. Figure 5 shows the light transfer vectors of several environment lights

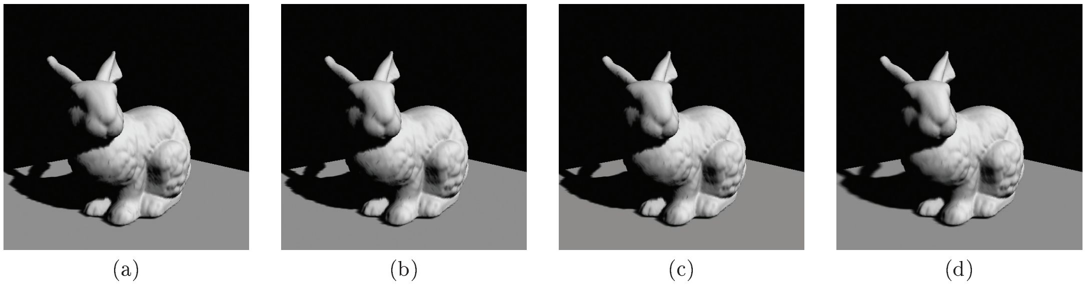

(a) (b) (c) (d)

Fig. 5 Visualizing (a) - (c) the light transfer vectors of several environment lights and (d) their corresponding clustered vector.

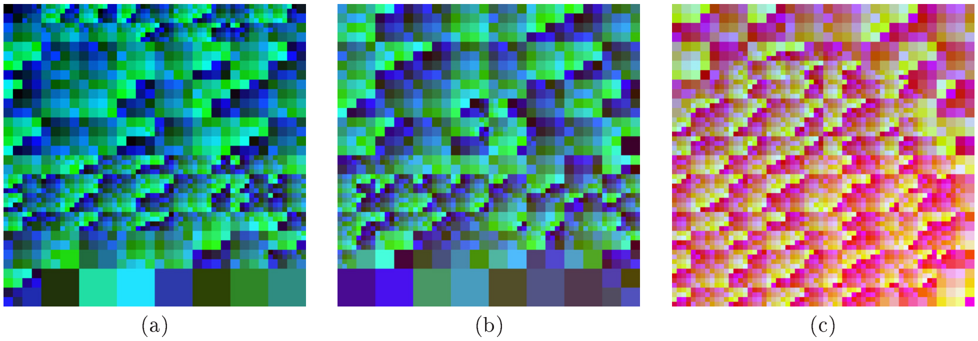

(a) (b) (c)

Fig. 6 The clustering results of the environment lights at the (a) front, (b) right, and (c) top faces of the environment illumination at resolution 1024 for the Bunny scene.

and the vector of their corresponding cluster. We can see that our method clustered the environment lights whose light transfer vectors are similar. Figure 6 shows the clustering results at several faces of the environment illumination for the Bunny scene. Each square represents a cluster of environment lights.

The examples of the relighting results are shown in Figures 7 and 8. It is clear that our method can produce both hard and soft shadows, which are the effects of all-frequency relighting. In particular, Figure 7(a) shows that our method can produce high-frequency shadows. In this example, there is only one small area light source located at the top left of the environment. In the glossy case, we can see highlights at the head, the belly, and the plinth of the Happy Buddha statue (see Figure 8(a)). In addition, the floor in Figure 8(b) has color variations, including highlights, which differs from the shading of the floor in the diffuse case.

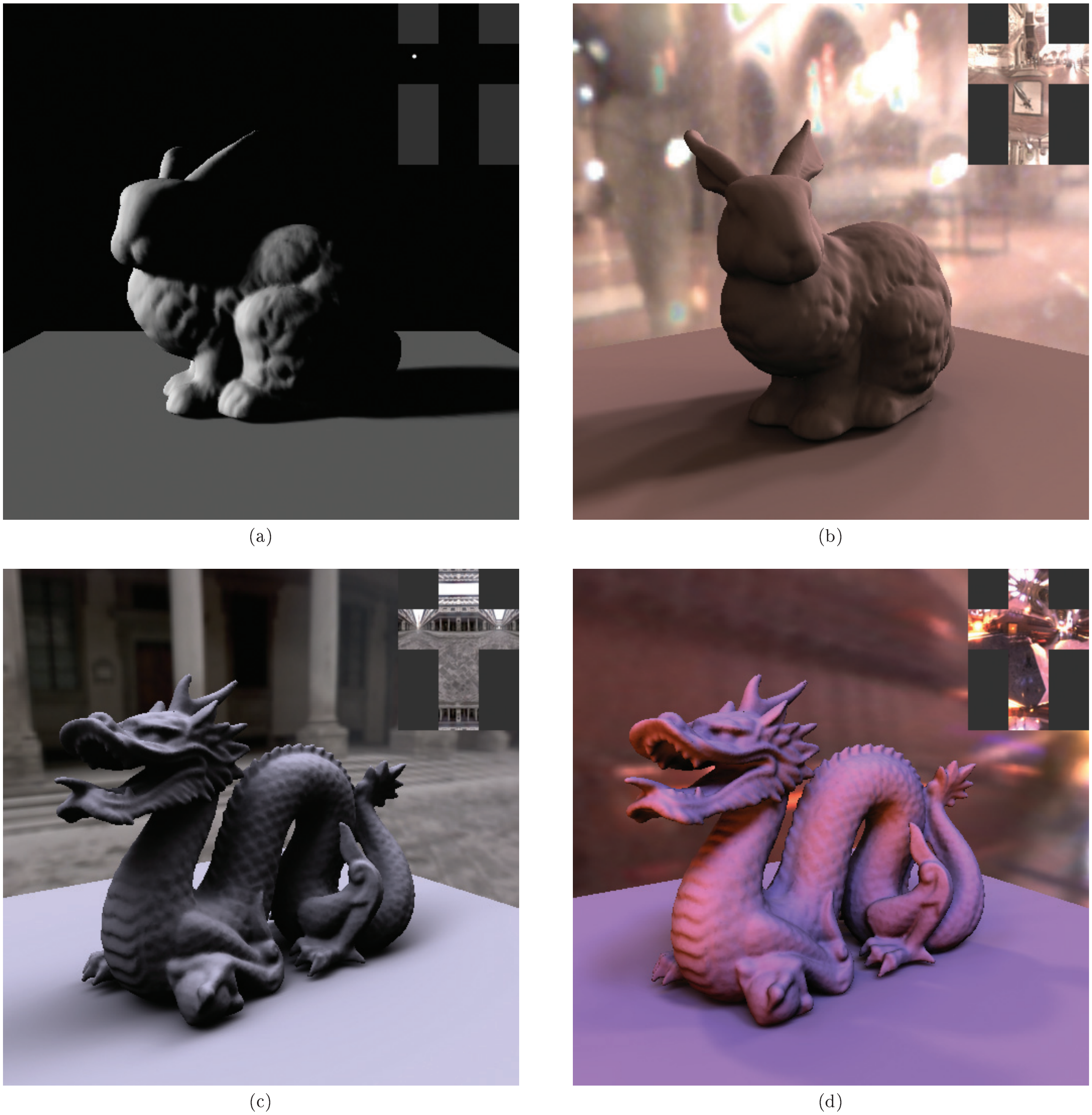
6.1 Changing the threshold value ϵ

We also performed experiments by changing the threshold value ϵ used in the clustering (see Section 4.3). In these experiments, the resolution of the environment illumination is 1024. In practice, we set ϵ to 5.0×10^{-5} and found that this threshold was sufficient for the production of high quality rendering results. Figure 9(c) shows that the result using our method exhibits a visual quality comparable to the result of exact integration due to all the environment lights.

When we increase ϵ, the environment illumination is approximated using a small number of clusters. In this case, either only shadows with blurred boundaries can be created or the resulting shadows are incorrect. Figure 9(d) shows the result when we set ϵ to 1.0×10^{-4}. In this case, the number of clusters is 1023. We can see that the shadow boundaries are blurred and that there are differences when compared to the reference image near the shadow boundaries (see Figure 9(e)). The shadow boundaries are blurred because many light transfer vectors, which are reasonably different, are clustered together. As a result, their light transfer functions are averaged and thus high-frequency shadows cannot be created. When we further increase ϵ to 2.5×10^{-4}, the number of clusters is decreased to 63 and the resulting shadows are incorrect (see Figures 9(f) and (g)). When we reduce ϵ, the number of clusters increases, resulting in a slower relighting.

6.2 Local changes in the environment illumination

For the Bunny scene, we conducted an experiment of relighting when some parts of the environment map had been manipulated (see Figure 10). In this experiment, the scene was initially illuminated using the beach environment map. Then, two circle light sources were added to the environment map, one at the front and one at the side of the Bunny. The radius of the two light sources were approximately 5% and 10% of the resolution of the environment illumination. We controlled the



(a) (b) (c) (d)

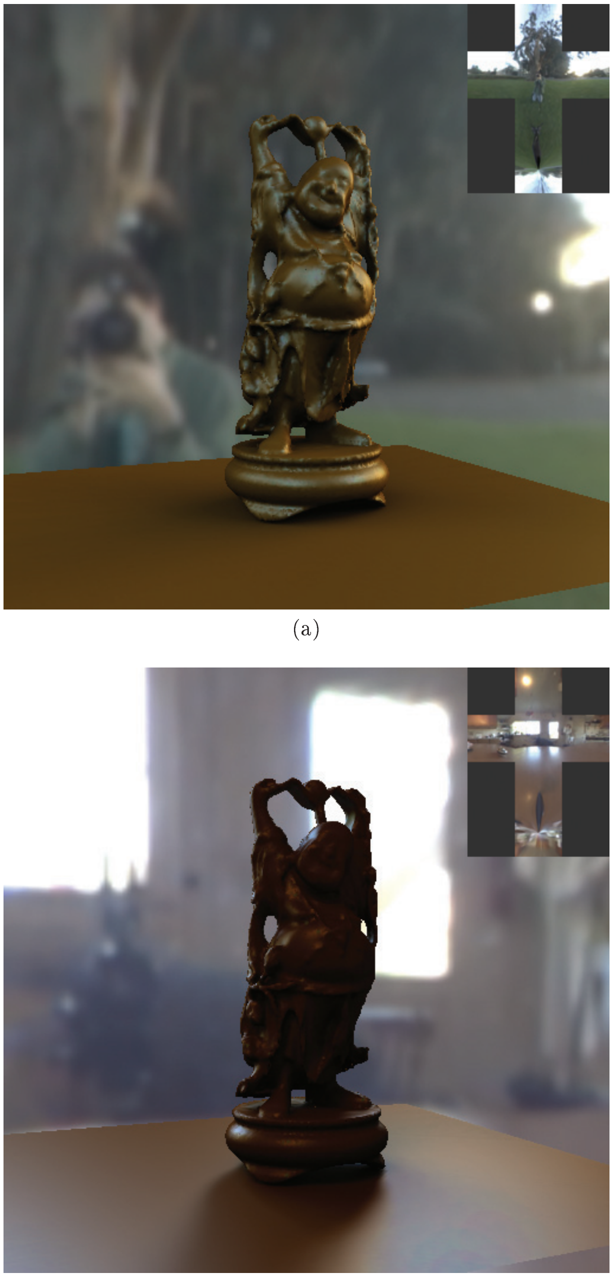
Fig. 7 Relighting a diffuse Stanford Bunny with (a) a small area light source and (b) the Galileo tomb environment illuminations. Relighting a diffuse Dragon model with (c) the Uffizi and (d) the Grace environment illuminations.

intensities of the light sources such that their intensities gradually increased. Later, the two light sources were moved. When the environment map was changed, using the method described in Section 5.2, it only took on average 0.06 sec for relighting. This is almost 100 times faster than the time to perform full recomputation.

6.3 Sub-linear property of the proposed algorithm

We performed tests, clustering the lights for the environment illumination at several resolutions for the Bunny scene. We used two values of ϵ in these tests. Table 2 shows the number of resulting clusters at resolutions $R = 256, 1024, 4096$.

We can see that even though the number of environment lights increase by 16 times (increasing R from 256 to 1024 and 1024 to 4096), the number of clusters increase only slightly (that is, a sub-linear increase). This sub-linear property is due to the fact that when we increase the resolution of the environment illumination, there is only an increase in the number of lights

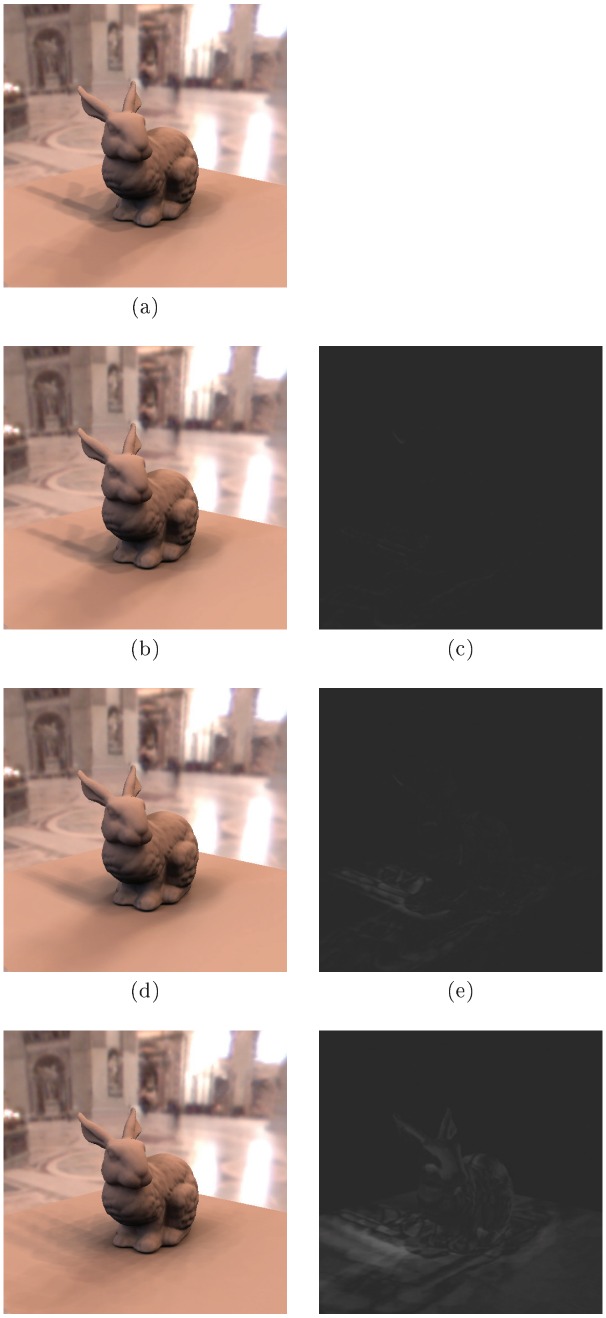

(a)

(b)

Fig. 8 Relighting the glossy Buddha and floor models with (a) a campus and (b) a kitchen environment illuminations.

with similar directions, and thus their light transfer vectors will be similar, resulting in only a small increase in the number of clusters.

Due to the sub-linear property, the proposed method can be applied for efficient relighting under very high-resolution environment maps. For instance, when $R = 4096$ and $\epsilon = 4.0 \times 10^{-5}$ (the number of clusters is 21393), our method can relight the scene in about 28 seconds. Note that the rendering results when setting ϵ to 5.0×10^{-5} and 4.0×10^{-5} have the same visual quality.


(a)

(b) (c)

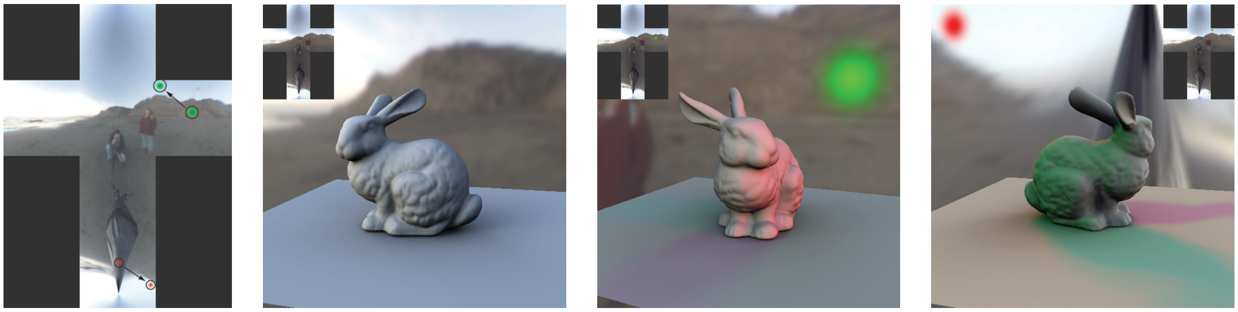
(d) (e)

(f) (g)

Fig. 9 Comparison between the results of the proposed method, by changing the threshold value ϵ, and the reference image created by exact integration under the St. Peters Basilica illumination at resolution 1024. (a) The reference image. (b) $\epsilon = 5.0 \times 10^{-5}$ (9831 clusters), (c) differences between (a) and (b). (d) $\epsilon = 1.0 \times 10^{-4}$ (1023 clusters), (e) differences between (a) and (d). (f) $\epsilon = 2.5 \times 10^{-4}$ (63 clusters), (g) differences between (a) and (f). Note that the differences shown in (c), (e), and (g) are emphasized.

6.4 Comparison to other clustering approaches

Table 3 shows the comparison between our method and other methods that employ clustering approaches. In



(a) (b) (c) (d)

Fig. 10 Manipulation of some parts of the environment illumination by (a) adding and moving light sources to the beach illumination, (b) initial rendering with the beach illumination, (c) adding a red area light source at the side and a green area light source at the back of the Bunny, and (d) moving the red and green area light sources.

Table 2 Clustering results on varying the resolution R of the environment illumination and the threshold ϵ for the Bunny scene.

Res. R	Lights	Clusters $\epsilon = 5.0 \times 10^{-5}$	Clusters $\epsilon = 4.0 \times 10^{-5}$
256	393216	8832	14736
1024	6291456	9831	19365
4096	100663296	10170	21393

Table 3 Comparison between the proposed method and other clustering approaches (Kristensen *et al.* [8] and Walter *et al.* [18]).

	[8]	[18]	Ours
Clustering independent to light intensities	√	-	√
Millions of lights	-	√	√
All-frequency shadows	-	√	√
Interactive / real-time rendering	√	-	√
Dynamic scenes	-	√	-
Changing viewpoint (arbitrary BRDFs)	√	√	-

comparison with the other two methods, ours is the only method that can interactively relight a scene using millions of lights and produce all-frequency shadows in the rendered images. However, our method has limitations, in that it can only handle rigid scenes and a fixed viewpoint for scenes with arbitrary BRDFs. Another limitation is that when a scene consists of many diffuse surfaces and a few glossy surfaces, there is a possibility that specular highlights cannot be produced accurately. When the scene consists of many diffuse surfaces, the clustering result is greatly influenced by the diffuse surfaces. The light transfer functions at diffuse surfaces tend to vary slowly between the neighboring environment lights whereas glossy surfaces do not have these tendencies. Therefore, during the clustering, even when the clustering condition (Equation 11 in Section 4.3) is satisfied, there is a chance that the clustering result cannot accurately capture the changes in transfer functions at the glossy surfaces.

7. Conclusions and Future Work

In this paper, we have presented an efficient approach for all-frequency relighting under environment illumination by clustering the environment lights. Since increasing the resolution of the environment illumination only increases the number of lights with similar directions, the increase in the number of clusters is very small compared with the increase in the resolution. As a result, our approach is able to relight under very high-resolution environment illumination. The important feature of our approach to using clusters is that for application in dynamic lighting environment, that is when some parts of the environment map are manipulated, it is easy to locate the affected clusters. Therefore, it is possible to achieve an efficient relighting by updating the radiance of only the affected clusters.

The future challenge is to extend the proposed method to permit the changing of the viewpoint when relighting a scene consisting of objects with arbitrary BRDFs.

Acknowledgments

We would like to thank Prof. Paul Debevec for the light probe images [2]. We would also like to thank the Stanford Computer Graphics Laboratory for the Bunny, Dragon, and Happy Buddha models. This work is partially supported by the Japan Society for the Promotion of Science.

References

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. *ACM Transactions on Graphics*, 22(3):605–612, 2003.
[2] P. Debevec. http://www.debevec.org/Probes.
[3] P. Debevec, T. Hawkins, C. Tchou, H. P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. *Proc. SIGGRAPH 2000*, pp.145–156, 2000.
[4] Y. Dobashi, K. Kaneda, H. Yamashita, and T. Nishita. A

quick rendering method for outdoor scenes using sky light luminance functions expressed with basis functions. *The Journal of the Institute of Image Electronics Engineers of Japan*, 24(3):196–205, 1995.

[5] J. Kautz and M. D. McCool. Interactive rendering with arbitrary BRDFs using separable approximations. *Proc. Eurographics Workshop on Rendering 1999*, pp.281–292, 1999.

[6] J. Kautz, P. P. Sloan, and J. Snyder. Shading and shadows: Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. *Proc. Eurographics Workshop on Rendering 2002*, pp.291–296, 2002.

[7] T. Kollig and A. Keller. Efficient illumination by high dynamic range images. *Proc. Eurographics Workshop on Rendering 2003*, pp.45–50, 2003.

[8] A. W. Kristensen, T. A. Moller, and H. W. Jensen. Precomputed local radiance transfer for real-time lighting design. *ACM Transactions on Graphics*, 24(3):1208–1215, 2005.

[9] J. Lehtinen and J. Kautz. Matrix radiance transfer. *Proc. Symposium on Interactive 3D Graphics 2003*, pp.59–64, 2003.

[10] X. Liu, P. P. Sloan, H. Y. Shum, and J. Snyder. All-frequency precomputed radiance transfer for glossy objects. *Proc. Eurographics Symposium on Rendering 2004*, pp.337–344, 2004.

[11] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics*, 22(3):376–381, 2003.

[12] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics*, 23(3):477–487, 2004.

[13] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. *Proc. SIGGRAPH 2001*, pp.497–500, 2001.

[14] R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. *ACM Transactions on Graphics*, 21(3):517–526, 2002.

[15] P. P. Sloan, J. Hall, J. Hart, and J. Snyder. Clustered principal components for precomputed radiance transfer. *ACM Transactions on Graphics*, 22(3):382–391, 2003.

[16] P. P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics*, 21(3):527–536, 2002.

[17] P. P. Sloan, X. Liu, H.-Y. Shum, and J. Snyder. Bi-scale radiance transfer. *ACM Transactions on Graphics*, 22(3):370–375, 2003.

[18] B. Walter, S. Fernandez, A. Arbree, K. Bala, M. Donikian, and D. P. Greenberg. Lightcuts: A Scalable Approach to Illumination. *ACM Transactions on Graphics*, 24(3):1098–1107, 2005.

[19] R. Wang, J. Tran, and D. Luebke. All-frequency relighting of non-diffuse objects using separable BRDF approximation. *Proc. Eurographics Symposium on Rendering 2004*, pp.345–354, 2004.

[20] R. Wang, J. Tran, and D. Luebke. All-frequency interactive relighting of translucent objects. *ACM Transactions on Graphics*, 24(3):1202–1207, 2005.

Henry Johan is an assistant professor in the School of Computer Engineering at Nanyang Technological University (Singapore) since 2006. He received his BS, MS, and Ph.D degrees in computer science from the University of Tokyo (Japan) in 1999, 2001, and 2004, respectively. From 2004 to 2006, he was a post-doctoral fellow in the Department of Complexity Science and Engineering at the University of Tokyo. His research interests include various areas of computer graphics, such as photorealistic and non-photorealistic rendering, shape blending, modeling, and image processing.

Tomoyuki Nishita is a professor in the Department of Complexity Science and Engineering (also in the Department of Information Science) at the University of Tokyo (Japan) since 1998. He received his BE, ME and Ph.D in engineering in 1971, 1973, and 1985, respectively, from Hiroshima University. He taught at Fukuyama University from 1979 to 1998. He was an associate researcher in the Engineering Computer Graphics Laboratory at Brigham Young University from 1988 to 1989. In 2005, he received the Steven A. Coons Awards from ACM SIGGRAPH. His research interests center in computer graphics including lighting models, hidden-surface removal, antialiasing, and natural phenomena.