社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2005－CG－120 (10)
2005／8／19

環境光源のクラスタリングによる効率的なリライティング

ヘンリー・ジョハン† 西田 友是†

本論文では，様々な全周波数の環境照明下での効率的な照明変化（リライティング）を行うための手法を提案する．環境照明は環境光源の集合として表現され，各環境光源は方向および輝度を持つ．前処理では，方向を考慮して環境光源を幾つかのクラスタに分類する．元の光源の数よりも環境光源を少数のクラスタに分類できることを実験によって確かめた．環境マップで表された環境光源の輝度を与えたら，前処理で求めたクラスタを用いて効率的なリライティングが可能となる．また，従来法よりも高解像度の環境照明にも対応できる．さらに，従来法と違って環境照明の一部が変更された際，提案法は効率よくリライティングを行うことが可能である．

Efficient Relighting by Clustering Environment Lights

HENRY JOHAN† and TOMOYUKI NISHITA†

We present a novel pre-computed radiance transfer method for an efficient relighting under arbitrary, all-frequency environment illumination. Environment illumination is represented as a set of environment lights. Each environment light has a direction and an intensity. In the pre-processing step, the environment lights are clustered into several clusters taking into account only their directions. Through experiments, we confirmed that the environment lights can be clustered into a small number of clusters compare to the original number of environment lights. Given any environment illumination sampled as an environment map, an efficient relighting is then achieved by computing the radiance using the pre-computed clusters. The proposed method also enables relighting under very high-resolution environment illuminations, resolution that is higher than any previous methods can handle. In addition, different from previous approaches, the proposed method can efficiently perform relighting when some regions of the given environment illumination changed.

1. Introduction

Producing photo-realistic images is one of the most important goal in computer graphics. Recently, rendering photo-realisitic images under environment illumination has gained much attention. However, in the case of environment illumination, we consider lights coming from all directions resulting in a high rendering cost when using conventional methods such as ray tracing. This becomes a serious problem, for instance in application such as lighting design where the lighting environment is continuously manipulated and rendering has to be performed each time.

With the emergence of the pre-computed 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 some basis functions, such as spherical harmonics[14] or wavelets[9),10]. The radiance at each vertex is then computed through these basis functions.

† 東京大学
The University of Tokyo

The limitation of using the spherical harmonics is that they can only be applied to low frequency lighting environment. Using wavelets overcome this problem, realizing high-frequency shadows in the rendered images. However, due to the computational cost and the storage requirements, they can only deal with environment illumination, which is represented as cubical environment map, up to resolution $6 \times 256 \times 256$. Another limitation is that full recomputation of relighting is required even when only some portions of the environment illumination changed.

We present a method for fast relighting that overcomes the two limitations mentioned above. Our method can deal with all-frequency lighting environment and thus we can produce all-frequency effects such as hard and soft shadows in the rendered images. If the scene consists of only diffuse objects, the view point can be changed during the relighting. For the scene consists of objects with arbitrary BRDF, the view point is fixed.

Different from 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 observed that environment lights whose directions are close to each other tend to have similar light transfer vector. Using this fact, we cluster the environment lights based on the similarities of their light transfer vectors. We will show that the environment lights can be clustered into a small number of clusters compare to the original number of environment lights while preserving the all-frequency effects in the rendered images. Given an environment map as the environment illumination, using the pre-computed clusters, an efficient relighting is achieved.

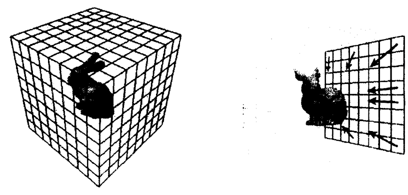
Our approach of clustering the environment lights also has the following advantages.

- Since increasing the resolution of the environment illumination is simply increasing the number of lights whose directions are close to each other, the increase in resolution results only in a relatively small increase in the number of clusters. As a result, relighting under very high-resolution environment illumination is possible.
- When the light intensities at some 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.*[3] presented a method for rendering under skylight illumination using basis functions. Ramamoorthi and Hanrahan[11),12] 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.*[14] proposed a pre-computed 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.*[5] proposed a method for rendering scenes with arbitrary BRDF. Lehtinen and Kautz[7] presented a method for rendering glossy objects efficiently. Sloan *et al.* used clustered principle component analysis for compressing the pre-computed data[13] and also presented a method to handle meso-structures on surfaces[15]. The limitation of these approaches is that they can only deal with low-frequency lighting environment.

Ng *et al.*[9),10] used wavelets instead of spherical harmonics and achieved relighting under all-frequency environment illumination. By using separable BRDF approximation[4], Wang *et al.*[16] and Liu *et al.*[8] extended the work in[9] to handle objects with complex BRDF. These approaches, however, can only handle environment illumination up to a certain moderate resolution. The highest resolution is $6 \times 256 \times 256$ achieved by the method in[10]. Another limitation is that even when only some parts of the environment illumination changed, these approaches have to perform full recomputation for the relighting.



(a) (b)

図 1 (a) Relighting a scene under environment illumination sampled as lights coming from a cube. (b) Several environment lights.

Agarwal *et al.*[1], Kollig and Keller[6] presented methods to accelerate rendering by reducing the environment lighting to hundreds of directional light sources. However, this method still take a few minutes to render an image. Moreover, given any environment maps, these methods have to convert the environment maps to a set of directional lights before the rendering can be performed. Our method, on the other hand, clusters environment lights based only on their directions, and thus the clustering results can be applied to any environment maps. 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 with resolution $6 \times R \times R$, where R is an arbitrary positive integer (Figure 1(a)). In other words, each face of the cube is subdivided into $R \times R$ regions. The directions of the environment lights are determined as the directions from the center of the regions on the faces of the cube to the center of the cube (Figure 1(b)).

3.1 Radiance due to environment illumination

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

$$B(x_i, \theta_i) = \sum_{j=1}^{M} L(\omega_j)\, V(x_i, \omega_j)\, \rho(x_i, \omega_j, \theta_i) \max(\omega_j \cdot \mathbf{n}(x_i), 0). \quad (1)$$

θ_i is the viewing direction, ω_j ($j = 1, ..., M = 6R^2$) are the incident light directions, $L(\omega_j)$ is the incident light intensity from direction ω_j, $V(x_i, \omega_j)$ is the binary visibility function indicating if the light from direction ω_j reachs x_i, $\mathbf{n}(x_i)$ and $\rho(x_i, \omega_j, \theta_i)$ are the normal vector and the BRDF at x_i, respectively.

Similar to Ng *et al.*[9], we consider two cases.

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

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

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

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

$$T_j(x_i) = V(x_i, \omega_j)\, \rho(x_i, \omega_j, \theta_i)\, \max(\omega_j \cdot \mathbf{n}(x_i), 0). \quad (3)$$

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

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

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

$$\mathbf{B} = T\mathbf{L}, \quad (5)$$

where $\mathbf{B}$ is a N-dimensional vector of radiance of all the vertices, T is a $N \times M$-transfer matrix, and $\mathbf{L}$ is a M-dimensional vector of 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.*[9] perform the Haar wavelet transform to each row of matrix 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 resulting in the increases of both the computational cost and the storage requirements of the precomputed data, which are proportional to the increase of R. This limits their approach to handle 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 a new environment map was given.

We propose an efficient method to compute the radiance at the vertices. In addition, our method overcomes the above-mentioned limitations. Instead of the row of T, we pay attention to the column of T. The j-th column of T represents the transfer function of each vertex with respect to the j-th environment light. In this paper, we will call the j-th column of 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

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

The cost for computing Equation (7) is reduced as follows. We pay attention to the fact that when the incident directions of several environment lights are close to each other, then there is a high probability that their light transfer vectors are 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 are approximated as

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

L_{C_k} is the radiosity 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 transfer vectors, all-frequency effects can be preserved in the rendered images.

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, in the relighting process, we can apply the clustering results to any given environment maps.

3.3 Overview of the proposed method

The proposed method has two steps.

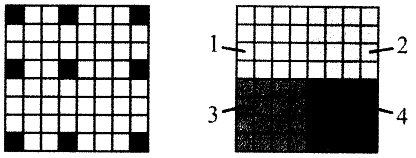
(1) Pre-processing: Given R, the environment lights in the environment illumination of resolution $6 \times R \times R$ are clustered based on their light transfer vectors. For each cluster C_k, its light transfer vector $\mathbf{T}_{C_k}$ is also computed.

(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 cluster the environment lights is to compute the light transfer vectors of all the environment lights and then performing the clustering using all the light transfer vectors. However, this approach is not efficient especially when the resolution of the environment illumination is high because the computational cost for 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, in our approach, we only choose some samples out of the whole environment lights and perform the clustering based on these samples.



(a) (b)

図 2 (a) Selecting samples of environment lights from the clustering domain. (b) Quadrisecting the region when the approximation error is large.

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 some environment lights as samples from the clustering domain (Section 4.1).
(2) Compute the light transfer vectors at the samples (Section 4.2).
(3) If these light transfer vectors are similar to each other, then all 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 small number of samples uniformly inside the domain. Basically, we choose nine samples of the environment lights. In order to cover all the possibilities of the light transfer vectors, we choose four samples at the corners, four samples at the middle of the edges, and one sample at the center of the domain (see Figure 2(a)). If the domain has less than nine samples, then we use all the environment lights inside the domain as samples.

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). During the clustering, several environment lights might be chosen as samples several times. In this case, it is not efficient if their light transfer vectors are computed several 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 not going to be chosen as sample any more, that is the environment light was classified to a specific cluster, its light transfer vector is deleted.

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.

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

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

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

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 to all samples $l \in S$.

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

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 and $\mathbf{T}_C$ as the approximated light transfer vector. Otherwise, we quadrisect the current clustering domain (see Figure 2(b)) and process each new domain independently.

5. Relighting

Assume that the environment lights of 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 radiosity 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 changed which means that the radiosities of some clusters changed. Let C_t $(1 \leq t \leq m)$ be one of such clusters and its radiosity changed from L_{C_t} to L'_{C_t}. The new radiances at all vertices $\mathbf{B}'_A$ due to the change in the radiosity of cluster C_t can be computed efficiently as follows.

The radiances at the vertices before the change are

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

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

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

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

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

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

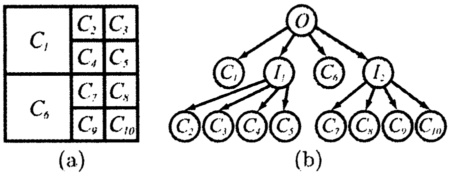


図 3 (a) An example of clustering result at one face of the cube and (b) its clusters distribution represented using a quadtree. O is the root, I represent the internal nodes, and C represent the leaf nodes which are clusters.

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

	Bunny	Dragon	Buddha
#vertices	73196	137499	69577
#res. R of env. illum.	1024	1024	256
#clusters	9831	5337	4701
#pre-comp. data (MB)	405	326	623
#pre-comp. time (min)	50	35	8
#relighting time (sec)	6.5	8	3.5

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

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

The only problem left is how to find the clusters whose radiosities changed when some regions of the given environment map 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 3). 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 radiosities 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.

6. Results

We performed the experiments using three scenes, a Stanford Bunny scene, a Dragon scene, and a Buddha scene. The Bunny and Dragon scenes consist of diffuse objects while the Buddha scene consist of specular objects. The statistics of these scenes are shown in Table 1. For compression, we quantize each element of the light transfer vector to one byte and discard all the zero elements. We used the environment map images provided by Paul Debevec[2] in our experiments. The computation were performed on a machine with Pentium 4 3.4 GHz.

Figures 4 and 5 show some examples of the relighting results. It is clear that our method enable to produce hard and soft shadows which are the

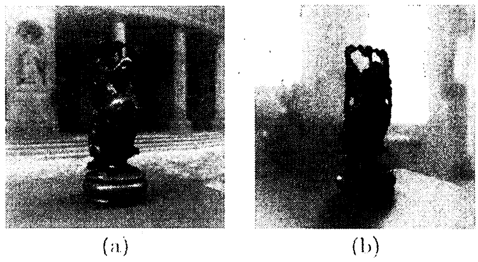


図 5 Relighting specular Buddha and floor models under (a) the Uffizi and (b) a kitchen environment illuminations.

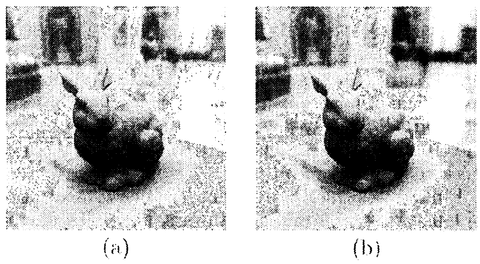
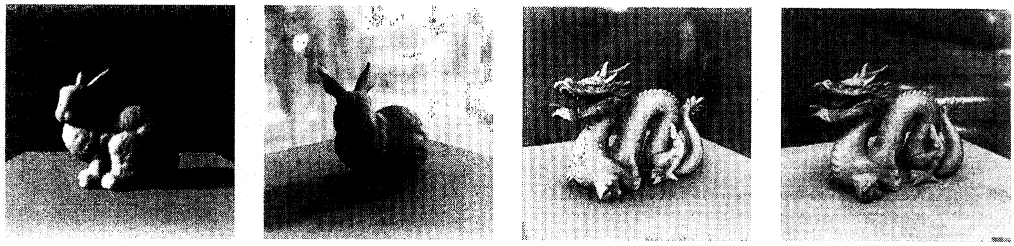


図 6 Comparison between the results of (a) our method and (b) the reference image created by exact integration under the St. Peters Basilica illumination.

effects due to all-frequency relighting. In particular, Figure 4(a) shows that our method enables to produce high frequency shadows. In this example, there is only one small area light source located at the top left in the environment. For the specular case, we can see highlights at the head, the stomach, and the stander of the Happy Buddha statue (Figure 5(a)). In addition, the floor in Figure 5(b) has color variations including highlights which is different to the shading of the floor in diffuse case.

We also performed experiments by changing the threshold value ϵ used in the clustering. For all the examples in this paper, we set ϵ to 5×10^{-5} and found that this threshold is sufficient for producing high quality rendering results. Figure 6 shows that the result of our method has a comparable quality to the result of exact integration due to all environment lights. When we increase the ϵ, then the environment illumination is approximated using a small number of clusters. In this case, we can only produce hard shadows. When we reduce the ϵ, then the number of clusters is increase resulting in a slower relighting.

For the bunny scene, we conducted experiment of relighting when some parts of the environment map are manipulated. In this experiment, the scene is initially illuminated using the beach environment map. Then, two circle light sources are added to the environment map, one at the front and one at the side of the bunny. The radius of the two light sources are around 5% and 10% of the resolution of the environment illumination. We control the



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

図 4 Relighting a diffuse Stanford Bunny under (a) a small area light source and (b) the Galileo tomb environment illuminations. Relighting a diffuse Dragon model under (c) the beach and (d) the Grace environment illuminations.

intensities of the light sources such that their intensities gradually increase. After that, the two light sources are 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 to perform full recomputation.

Finally, for the bunny scene, we performed a test of clustering environment illumination with resolution $6 \times 4096 \times 4096$. For this resolution, the number of resulting clusters is 10170 clusters. Compare to the case of resolution $6 \times 1024 \times 1024$, even tough the number of environment lights increase 16 times, the number of clusters increase only slightly (9831 to 10170). This fact shows that the proposed method can be applied for efficient relighting under very high-resolution environment maps.

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 cluster is relatively small compare to the increase in the resolution. As a result, our approach is able to relight under very high-resolution environment. In addition, clustering can be efficiently performed by processing only a set of light samples. Another advantage of using clusters is that for application such as lighting design, when some parts of the environment map are manipulated, it is easy to find the affected clusters. Thus, efficient relighting is possible by updating the radiance due only to the affected clusters.

The future challenge is to extend the proposed method for allowing changing of view point when relighting a scene consisting of objects with arbitrary BRDF.

参 考 文 献

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, July 2003.
2) P. Debevec. http://www.debevec.org/Probes.
3) 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, July 1995.
4) J. Kautz and M. D. McCool. Interactive rendering with arbitrary BRDFs using separable approximations. *Proc. Eurographics Workshop on Rendering 1999*, pp.281–292, June 1999.
5) 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.
6) T. Kollig and A. Keller. Efficient illumination by high dynamic range images. *Proc. Eurographics Workshop on Rendering 2003*, pp.45–50, 2003.
7) J. Lehtinen and J. Kautz. Matrix radiance transfer. *Proc. Symposium on Interactive 3D Graphics 2003*, pp.59–64, 2003.
8) 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.
9) 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.
10) R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics*, 23(3):477–487, 2004.
11) R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. *Proc. SIGGRAPH 2001*, pp.497–500, 2001.
12) R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. *ACM Transactions on Graphics*, 21(3):517–526, 2002.
13) 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, July 2003.
14) 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.
15) P.-P. Sloan, X. Liu, H.-Y. Shum, and J. Snyder. Bi-scale radiance transfer. *ACM Transactions on Graphics*, 22(3):370–375, July 2003.
16) 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.