Isocube-based Spherical Wavelet for Image-based Relighting

Liang Wan[†,‡], Chi-Sing Leung[†], Ping-Man Lam[†], Shue Kwan Mak[†] and Tien-Tsin Wong[§]

[†] Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong

[‡] School of Computer Software, Tianjin University, Weijin Avenue 92#, Tianjin, P.R. China

[§] Department of Computer Science and Engineering, the Chinese University of Hong Kong, Shatin, Hong Kong

Abstract

Using the brute-force way to synthesize a scene that is illuminated under a distant environment is computationally intensive. To speed up the synthesis, we can represent the appearance of the scene under different lighting directions as an illumination adjustable image (IAI). Relighting the scene under a lighting environment is then achieved by manipulating the pre-recorded images of the IAI. However, the relighting based on the pre-recorded images is still inefficient due to the large storage and computation requirements. The approaches based on spherical harmonics (SH) and cubemap-based spherical wavelets (CSW) have been proposed to improve the relighting efficiency. Note that the SH approach is suitable for low-frequency lighting only, while the CSW approach suffers the problem of uneven sampling. In this paper, we propose a novel approach, Isocube-based spherical wavelets (IsoSW), to efficiently synthesize images based on the IAI representation. Unlike the SH approach, our approach can effectively capture high-frequency illumination effects. Compared with the CSW approach, our approach not only improves the rendering quality, but also makes the synthesis process more efficient.

Index Terms

Image-based relighting, illumination adjustable image, Isocube mapping, spherical wavelets

I. Introduction

Synthesizing images of an object/scene under a distant lighting environment has been extensively studied in recent years [1][2][3]. To enrich the rendering realism, illuminations captured in natural environments are widely used as the lighting environment, which is modelled as a spherical function. The brute-force way that integrates all the contributions from the lighting environment is computationally intensive. Although we can apply importance sampling techniques [4] to simplify the integration process, the synthesis process is still time consuming for a complex 3D scene. One successful way to improve the speed of synthesizing images is to use the image-based relighting (IBL) approach [5][6][7][8]. Unlike the traditional geometry-based rendering, the IBL approach synthesizes images of an object or a 3D scene from pre-recorded images. Its advantage is that the rendering time is independent of the scene complexity. Hence, interactive or real-time rendering could be achieved. In addition, complex lighting effects, such as shadowing, inter-reflection, refraction and other phenomena, can be faithfully generated using the IBL approach.

As a typical IBL technique, the illumination adjustable image (IAI) model [9][10] allows users to adjust the illumination. Its basic idea is to capture the appearances of a 3D scene under various lighting directions. In this representation, a pixel is not associated with a single radiance value only but with a number of sampled radiance values. An IAI of a 3D scene contains thousands of reference images, each image corresponding to a different lighting direction. Given an IAI, images under novel illumination conditions are generated by superimposing proper reference images. Since the quality of synthesized images depends on the number of reference images that are used, this approach has the disadvantage of the enormous storage requirement. Furthermore, the synthesis with a novel lighting environment requires the integration of all incoming radiances, which is time-consuming.

To tackle the problems associated with the IAI model, the spherical harmonics (SH) [11] has been proposed to represent the IAI data set [2][10]. Since the radiance values of a pixel constitute a spherical function, the SH approach represents the radiance values by a weighted sum of SH basis functions. Then each pixel in the IAI becomes a SH coefficient vector. The SH coefficients of the same order from different pixels can be organized as a SH lightmap. As a result, an IAI is represented by a limited number of SH lightmaps. This representation reduces the storage requirement for an IAI greatly. On the other hand, the lighting environment can also be represented by a SH coefficient vector. To synthesize the 3D scene under a novel lighting environment is to compute the dot-product of the SH lightmaps and the SH coefficient vector from the lighting environment. Although the SH approach improves the efficiency on both storage and rendering, they are only suitable to handle low-frequency lighting, since the SH representation discards the information of high-frequency lighting. Hence the SH approaches are weak in capturing high-frequency illumination features, such as shadows and highlights.

To capture high-frequency lighting effects, the cubemap-based spherical wavelet (CSW) approach was proposed [12]. In this approach, a spherical function is represented by a cubemap. By applying wavelet transform on the cubemap-based samples, each pixel in an IAI is associated with a vector of CSW coefficients. In other words, an IAI is represented by a number of CSW coefficient lightmaps. Since the lighting environment undergoes the similar CSW transform, the rendering process is still a dot-product process, but in terms of wavelet coefficients. Note that the CSW transform keeps all the information of the original data. Using all CSW coefficients is equivalent to the integration of all radiance values. Hence, to speed up the rendering and reduce the storage, only a part of important CSW coefficients are used in approximation.

However, using the cube mapping to represent a spherical function suffers the problem of uneven sampling, as the cube mapping samples more densely at cube face corners but less at cube face centers. This may result in a poor representation of lighting features from the environment illumination near face centers. Figure 1 shows the synthesized results of a teapot under the lighting environment that contains an area light source from above. In Figure 1(b), we uniformly sample the lighting environment with a large number of sampling points, which generates a high-quality rendering result. In Figure 1(c), we use a $6 \times 32 \times 32$ cubemap to represent the lighting environment and IAI data set. Note the CSW approach creates obvious under-sampling artifacts in the shadow when 200 most important coefficients are used. The artifacts can be suppressed if we increase the cubemap resolution and use more coefficients. However, such trivial solution will increase both the computational and storage complexities. Furthermore, due to uneven sampling, more wavelet coefficients are involved near face corners than those at face centers. Hence, unbalanced rendering quality will result. In other words, we have to use different number of coefficients in different regions in order to achieve more balanced rendering quality. This will make the synthesis process less efficient.

In this paper, based on the concept of even area spherical partitioning [15][16], we develop the Isocube-based spherical wavelet approach (IsoSW), which handles the disadvantages of the CSW approach successfully. The major drawback of the CSW approach is uneven sampling of spherical functions. If we sample spherical functions uniformly, we are able to improve the synthesis quality and to reduce the number of wavelet coefficients to be used.

In the literature, icosahedron (20-face polyhedrons) [13][14] is adopted as the base for spherical wavelet transform. Although it produces more uniform samples than the cube mapping, its triangular partition makes it less efficient to access the nearest samples. We may use a table lookup technique to improve the efficiency, but more storage may be required. There also exist several spherical mappings that have quadrilateral faces and uniformly partition the spherical surface, like HEALPix [15] and Isocube [16]. These mappings have been proven to be equal-area for all the resulted samples. Thus, we can safely ignore the contribution of sample's solid angle (spherical area), leading to more efficient computation.

The Isocube mapping [16] is inspired from the HEALPix mapping [15], both sharing similar uniformity. In comparison, the HEALPix mapping uses 12 base faces to cover the spherical surface but Isocube mapping uses 6 rectangular faces only. The discrete wavelet transform should be performed over each base face in the HEALPix mapping and Isocube mapping. We can consider the wavelet process is a piece-wise transform. When a mapping involves more base faces, the efficiency of the corresponding transform is reduced. For example, when more base faces are involved, more low frequencies are resulted. It is because for each base face, we need certain number of low frequency coefficients for the approximation. Hence, a mapping with more base faces may lead to a less efficient approximation. Therefore, we adopt the Isocube mapping to represent spherical functions in our approach.

Although the CSW approach has six faces only, it has the uneven sampling problem. That means, more wavelet coefficients are involved near face corners than those at face centers. In contrast to the CSW approach, our approach uses a similar number of wavelet coefficients for different regions of cube faces. Hence much more balanced rendering quality is achieved over the whole spherical domain. As shown in Figure 1(d), the synthesized result from our approach faithfully preserves the shadow information, quite similar to the control image. It has no under-sampling artifacts as the CSW approach does when 200 most important wavelet coefficients are used.

In the rest of the paper, we first introduce background of the illumination model and cube mapping in Section II-A. Section III presents the details of the IsoSW approach for relighting. Section IV shows the experimental results while Section V concludes our work.

II. Background

A. The Illumination Model

The IAI model adopts the plenoptic illumination function (PIF) [9][10] to describe the lighting property of a 3D scene. As a computational human vision model, the PIF describes the radiance value received along any viewing direction $\vec{v}$ arriving at any point $\dot{E}$ in space, over any range of wavelength λ. The PIF has the following formulation,

$$I = P(\vec{v}, \vec{\omega}, \dot{E}, \lambda), \tag{1}$$

where I is the radiance; $\dot{E} = (E_x, E_y, E_z)$ is the viewpoint; $\vec{v}$ specifies the viewing direction originated from the viewpoint $\dot{E}$; $\vec{\omega} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ specifies the direction of a directional light source illuminating the 3D scene.

In our discussion, we fix the viewpoint and approximate the contribution of different wavelengths with R/G/B combination. Then, the sampling process of PIF is a taking photo process at a fixed point. In this paper, we consider the image based relighting application. Hence, the viewpoint is fixed. Our approach can be extended to the variable viewpoint case. We expect that the performance difference between the CSW and IsoSW approaches under the variable viewpoint situation will be similar to that under the fixed viewpoint situation. It is because under the fixed viewpoint situation, the viewing directions of different elements of the 3D object are different.

Considering the perspective views, $\vec{v}$ actually specifies a pixel position (x, y) in the view-plane, as illustrated in Figure 2. Hence, the PIF has the short form,

$$I = P(x, y, \vec{\omega}). \tag{2}$$

Although the exact form of a PIF is usually unknown, sampling the PIF can be considered as a process of taking pictures. The simplified PIF is controlled by two sets of parameters,

(x, y) and $\vec{\omega}$. We can sample $\vec{\omega}$ on the unit sphere, and further take a picture of the 3D scene under each sampled lighting direction (see Figure 3). Consequently, the complete IAI consists of a set of reference images, $\{\mathbf{P}(\vec{\omega}_1), \cdots, \mathbf{P}(\vec{\omega}_M)\}$, illuminated under M various lighting directions, where $\mathbf{P}(\vec{\omega}_i) = P(x, y, \vec{\omega}_i)$. It should be noticed that a PIF can be reduced to a spherical function of $\vec{\omega}$ when we fix the pixel position (x, y).

Now, the question is how to sample the lighting direction $\vec{\omega}$. A simple way is to use spherical grid based sampling [9], which uniformly samples the elevation angle θ and azimuth angle ϕ. The disadvantage is that the sample points are not evenly distributed on the sphere. In this simple sampling scheme more samples are near the sphere poles. In the CSW approach [12], $\vec{\omega}$ is sampled on the grid of six cube faces of a cubemap. As discussed previously, the cube mapping does not sample the spherical surface uniformly. In our work, we adopt Isocube [16], which distributes the samples more evenly on the spherical surface (see Figure 3). Its details are presented in the next section.

With the sampled PIF, we can relight the scene with multiple directional light sources using the following formula [9][10]:

$$I(x, y) = \sum_{i=1}^{M} P(x, y, \vec{\omega}_i) L(\vec{\omega}_i), \tag{3}$$

where M is the total number of sampling light directions, $P(x, y, \vec{\omega}_i)$'s are the sampled PIF values obtained from the pixel position (x, y) of reference image $\mathbf{P}(\vec{\omega}_i)$, and $L(\vec{\omega}_i)$ describes the lighting environment. The function $L(\vec{\omega}_i)$ tells us the light intensity along the direction $\vec{\omega}_i$.

When the light source is a distant environment, we sample the lighting environment by a set of sampling points $\{\vec{\omega}_1, \cdots, \vec{\omega}_M\}$, and get a set of sampled lighting values $\{L(\vec{\omega}_1), \cdots, L(\vec{\omega}_M)\}$. These sampling points can be taken from Isocube mapping [16] or cube mapping [20].

Relighting the 3D scene becomes to integrate the contributions of the M lighting values and the corresponding IAI reference images. It is worth mentioning that the lighting environment is usually represented as a high dynamic range (HDR) environment map [17]. In contrast to the traditional low dynamic technique that provides 8-bit precision, the HDR technique, that usually uses the floating point format to represent the lighting intensity, has a greater dynamic range. HDR images can more accurately represent the wide range of intensity levels in the real world.

Obviously, it is time-consuming to directly compute the above integration, i.e., (3). To reduce the computational complexity, several transform-based approaches [2], [10], [12], [18] have been proposed. With an appropriate transform, only a part of transform coefficients, which have higher importance, are considered in the integration.

B. Cube Mapping

The cube Mapping [20] is the most popular spherical mapping in the graphics community. It can be considered as a technique to partition a spherical surface. With the cube mapping, a spherical function can be projected onto the six faces of a cube and stored as six square textures.

Consider a cube shown in Figure 4(a). It consists of six faces, namely cubemap faces. We can partition each face into a number square cells by two sets of parallel grid lines. Each set of parallel grid lines is of uniform intervals on the cubemap faces. As shown in Figure 4(b), the six cubemap faces can be unfolded into six regions of a 2D texture. The texture are called unrolled map. To partition a sphere, each face is projected onto a sphere, as shown in Figure 4(c). During projection, the grid lines are mapped to some geodesic curves on the sphere. Now, a sphere is partitioned into a number of rectilinear cells on the sphere. Based on the partitioning, we can digitize/sample a spherical function.

We consider the centers of cells as sampling points. Figure 4(d) shows the sampling points on the sphere. With the sampling points, we are able to represent a spherical function by six square images. Given a spherical function, we use the sampling points to sample the spherical function. Afterwards, we put the sampled values on the corresponding cells on the six cube faces. Hence we can obtain six square images. Figure 5(a) shows a simple spherical function. In this spherical function, only a region is with large values. After sampling, we can obtain six square images that store the sampled spherical function, as shown in 5(b).

C. Cubemap spherical wavelets

With the cube mapping, a sphere can be digitized into six square faces. Hence, we can utilize the conventional 2D Haar wavelet basis to define a spherical wavelet basis. That is, we define a Haar basis on each cubemap face. For example, if the resolution of a face is 4×4, we can define 16 basis functions on a face. Since there are six faces in a cubemap, there are 96 basis functions $\varphi_j^{cm}(\vec{\omega})$'s. Figure 6 shows some basis functions in the unrolled view.

With the wavelet basis functions, a spherical function $P(\vec{\omega})$ can be expressed as

$$P(\vec{\omega}) = \sum_{j=1}^{M} \lambda_j^{cm} \varphi_j^{cm}(\vec{\omega}) \tag{4}$$

where λ_j^{cm}'s are the wavelet coefficients, $M = 6 \times N \times N$ is the number of basis functions in the CSW basis, and N is the resolution of an axis in the cubemap faces. In [12], the CSW approach was proposed for relighting an IAI. However, using the cube mapping to represent a spherical function suffers the problem of uneven sampling, as shown in Figure 4(d).

In the CSW approach, the cube mapping samples more densely at cubemap face corners but less at cubemap face centers. Hence, the face center regions have a poor sampling ability, as shown in Figure 1. On the other hand, the quality of approximation in the face corner regions could be good. However, more wavelet coefficients are required involved because the sampling points are denser in the face corner regions. That means, in the face centers, edges, and corners have different sampling ability. Besides, they need different number of wavelet coefficients for approximation. The uneven sampling pattern in the CSW approach makes the relighting inefficient. Hence, using a uniformly sampling spherical wavelet for rendering is important.

III. Efficient Rendering with IsoSW Approximation

In this section, we will introduce our IsoSW approach. We will first discuss the way to partition a sphere based on the Isocube mapping [16], which handles the disadvantages of the cube mapping. Afterwards, we will present the IsoSW approach to relight an IAI.

A. Isocube mapping

In the cube mapping, we uniformly partition the faces of a cube into a number of cells. Afterwards, we project the partitioning onto the spherical surface. However, the partitioning is not uniform on the spherical surface. Also, those cells on the spherical surface are with different spherical areas.

The Isocube mapping [16] is an equal-area, six-face spherical mapping. The idea is that we directly performs partitioning on the spherical surface. The first-level partitioning produces 6 base faces. Recursive subdivision of the base faces obtains equal-area smaller elements. More specifically, the sphere is first divided into equatorial and polar zones by “arctic” and “antiarctic” circles at $z = \pm 2/3$, as shown in Figure 7(a). Since the area of the equatorial zone is four times that of each polar zone, the equatorial zone is subdivided into four symmetric regions, hence yielding 6 base faces, as shown in Figure 7(b).

Within the equatorial zone $(|z| \leq 2/3)$, the sphere is partitioned by curvilinear equations [16] in the form,

$$\phi = \frac{\pi}{2}\,(\frac{k}{N} - \frac{1}{2}), \quad k = 0, \ldots, 4N \tag{5}$$

indicated as blue curves in Figure 8, and

$$z = \frac{2}{3}\,(\frac{2l}{N} - 1), \qquad l = 0, \dots, N \tag{6}$$

indicated as magenta curves in Figure 8. Here, $z = \cos\theta$; the parameter N controls the subdivision level, which should be a positive integer. On the other hand, the polar zones ($|z| > 2/3$) are partitioned by another set of curves [16]:

$$\phi \;=\; \frac{\pi}{2}\frac{2k+\varepsilon}{2N}\cdot\frac{1}{\sqrt{3(1-z)}},\;\; k = 0, \dots, \left\lfloor\frac{N}{2}\right\rfloor - 1 \tag{7}$$

$$z \;=\; 1 - \frac{1}{3}\left(\frac{2l+\varepsilon}{N}\right)^2, \qquad l = 1, \dots, \left\lfloor\frac{N}{2}\right\rfloor\,. \tag{8}$$

The above equations are defined within the region where $(\phi, z) \in [\,0, \frac{\pi}{4}\,] \times [\frac{2}{3}, 1]$. They can be applied to other polar regions due to symmetry.

With (5)-(8), the Isocube mapping divides a spherical surface into six spherical faces, namely Isocube faces. Each Isocube face is a 2D array of cells. We called these cells *Isocube cells*. Note that Isocube cells are with the same spherical area. Similar to the cube mapping, we can consider the center of an Isocube cell as a sampling point. Figure 8(d) shows the sampling points of the Isocube mapping. From Figure 4(d) and Figure 8(d), we can observe that the sampling points of the Isocube mapping are more uniformly distributed on the sphere. We can use the Isocube sampling points to sample a spherical function. Afterwards, we can obtain six square images that store the sampled spherical function, as shown in Figure 9.

B. IsoSW Approximation

With the Isocube mapping, a sphere can be digitized into six square Isocube faces. Hence, similar to the CSW case, we are able to define a Haar wavelet basis on an Isocube face. Then, we can have a spherical wavelet basis based on the Isocube mapping. We call this approach ”Isocube spherical wavelet (IsoSW)”.

With the IsoSW basis, the lighting environment $L(\vec{\omega})$ can be expressed as

$$L(\vec{\omega}) = \sum_{k=1}^{M} \mu_k \varphi_k(\vec{\omega}), \tag{9}$$

where μ_k's are the IsoSW coefficients of the lighting environment, and $\varphi_k(\vec{\omega})$'s are the IsoSW basis functions.

For the IAI data set we prepare it the following way. When one sampling light direction $\vec{\omega}_i$ is specified, a reference image is synthesized. Figure 3 shows some images of our IAI data set. Each reference image contains the radiance values of all the pixels on the view-plane for a given lighting direction. Before applying the wavelet transform, we group together all the radiance values of a pixel under different sampling directions. These values actually form a spherical function (refer to the illustration in Figure 10).

To simplify the notation, we drop the position variables and write the spherical function $P(x, y, \vec{\omega})$ as $P(\vec{\omega})$. For each pixel, we are able to expand $P(\vec{\omega})$ in the wavelet basis, given by

$$P(\vec{\omega}) = \sum_{j=1}^{M} \lambda_j \varphi_j(\vec{\omega}) \tag{10}$$

where λ_j's are the IsoSW coefficients of $P(\vec{\omega})$.

It is well known that the orthonormal basis functions satisfy $\sum_{i=1}^{M} \varphi_j(\vec{\omega}_i)\varphi_k(\vec{\omega}_i) = \delta_{j,k}$, where Kronecker deltas $\delta_{j,k} = 1$, if $j = k$, otherwise $\delta_{j,k} = 0$. We now substitute Equations (9) and (10) into Equation (3) and get,

$$\begin{aligned} I &= \sum_{i=1}^{M} \left(\sum_{j=1}^{M} \lambda_j \varphi_j(\vec{\omega}_i) \right) \left(\sum_{k=1}^{M} \mu_k \varphi_k(\vec{\omega}_i) \right) \\ &= \sum_{j=1}^{M} \sum_{k=1}^{M} \lambda_j \mu_k \sum_{i=1}^{M} \varphi_j(\vec{\omega}_i)\varphi_k(\vec{\omega}_i) = \sum_{j=1}^{M} \sum_{k=1}^{M} \lambda_j \mu_k \delta_{jk} \\ &= \sum_{j=1}^{M} \lambda_j \, \mu_j \, . \end{aligned} \tag{11}$$

C. Lightmap and Important coefficients

After we perform the IsoSW transform to the IAI data set (see Equation (10)), the radiance values associated with a given pixel are transformed to an M-dimensional coefficient vector, $[\lambda_1, \cdots, \lambda_M]$. If we pick the first coefficients (the light gray elements in Figure 11) of all the coefficient vectors, they form an IsoSW lightmap. Similarly, the second coefficients (the darker gray elements) of all the coefficient vectors are grouped to form another lightmap. We apply this process to all the coefficients, and obtain a set of M lightmaps. As visualized in Figure 11, each lightmap is a floating point image. The lighting environment is also a spherical function. After performing the IsoSW transform to the lighting environment (see Equation (10)), we have M IsoSW coefficients, $[\mu_1, \cdots, \mu_M]$.

With the M Isocube lightmaps and the M IsoSW coefficients of the lighting environment, we may compute the relighting using Equation (11). It is noticed that Equation (11) uses two M-dimensional vectors. Its computational cost is equivalent to that of Equation (3), which is time-consuming. Recall that the wavelet transform redistributes the image energy such that in the transform domain, some coefficients are more important than the others, which means they have much stronger impacts on the final rendering results. Consequently, Equation (11) can be simplified by using a limited set of most important wavelet coefficients.

In the implementation, we choose the important coefficients based on the environment lighting. More specifically, we employ the area-weighted scheme [12], i.e. scaling the priorities of coefficient μ_k's by their corresponding spherical areas. Based on the area-weighted scheme in [12], the priority $\mathcal{P}_k$ of a coefficient μ_k is given by

$$\mathcal{P}_k = \mu_k \mathcal{A}_k \tag{12}$$

where $\mathcal{A}_k$ is the spherical area of the basis function $\varphi_k(\vec{\omega})$. Larger value of $\mathcal{P}_k$ represents the coefficient is more important. As mentioned in [12], the area-weighted scheme gives higher priorities to coefficients with larger areas and helps to resolve the diffuse component of the scene more quickly.

The IsoSW coefficients of the lighting environment are stored in a descending order based on the above priority. The re-ordered IsoSW coefficients are given by

$$[\mu_1, \cdots, \mu_M] \ \ \overrightarrow{\text{sorting}} \ \ = [\mu_{\pi_1}, \cdots, \mu_{\pi_M}\ ] \,. \tag{13}$$

During the rendering, only forefront (important) IsoSW coefficients of lighting environment are considered. Hence, the relighting equation (eq:isoSW) becomes

$$I \approx \sum_{j=1}^{M'} \lambda_{\pi_j} \, \mu_{\pi_j} \ , \tag{14}$$

where M' is usually much smaller than M.

D. Rendering with Dynamic Lighting Environment

In previous parts, we assume the lighting environment is static. When the lighting environment is time-varying, as modelled by $\{L(\vec{\omega}, t) \,|\, t = 1, \cdots, T\}$, we perform the rendering for each of environment frames separately. Since the illumination is not fixed, the important wavelet coefficients may be different for the frames. Therefore, we re-compute the priorities of coefficients for each frame $L(\vec{\omega}, t)$. Afterwards, we select the most important coefficients frame by frame. Note that during this process, the set of lightmaps is computed only once, while different lightmaps may be selected for different environment frames.

IV. Experiments and Discussion

In the experiments, we test our approach for two 3D models, TEAPOT and BUNNY. For each model, two IAI data sets are generated by sampling on the cube mapping and the Isocube mapping, both in the resolution of $6 \times 32 \times 32 = 6,144$. On the other hand, we generate a very dense data set with 81,920 uniformly-distributed sampling lighting directions.

This dense data set is used to synthesize control results. For the lighting environments, we prepare two HDR panorama sequences: GRACE and ST. PETER, by rotating the static HDR image [17]. Each environment sequence contains $T = 120$ frames (some frames are shown in Figure 12). Then we sample each environment frame at the 6,144 cube mapping and Isocube mapping sampling points.

A. Error in Lighting Approximation

In our approach, we approximate the lighting environment using a small amount of important wavelet coefficients. This inevitably introduces some errors. In other words, the less coefficients are used, the less accurate the reconstructed environment map will be. To evaluate the reconstruction accuracy, we measure the percentage error $E(t)$ of the reconstructed environment maps $\hat{L}(\vec{\omega}, t)$ [12], given by

$$E(t) = \frac{\sum_i (\hat{L}(\vec{\omega}_i, t) - L(\vec{\omega}_i, t))^2}{\sum_i (L(\vec{\omega}_i, t))^2}. \tag{15}$$

It is defined as the ratio of the summed square differences between $\hat{L}(\vec{\omega}, t)$ and the original data $L(\vec{\omega}, t)$ against the summed square of $L(\vec{\omega}, t)$. To account for the dynamic lighting environment, we compute the average percentage errors over all environment frames, i.e., $\bar{E} = \sum_t E(t)/T$.

Figure 13 the average percentage errors of the CSW and IsoSW approaches when different number of important wavelet coefficients are used. As shown in the figure, the shape of the two error curves of the two approaches is similar because the two approaches use the same selection method for important wavelet coefficients. When more coefficients are used for the approximation, the less percentage errors are obtained.

From the figure, the performance of the IsoSW approach is better than that of the CSW approach. For the “Grace” sequence, when 200 coefficients are used, the average error of the IsoSW approach is around 32 % only. But the average error of the CSW approach is 38%.

For the “ST. PETER” sequence, when 200 coefficients are used, the average error of the IsoSW approach is around 37 % only. But the average error of the CSW approach is 42%.

For the same average percentage error, the IsoSW approach needs less wavelet coefficients for approximating the lighting environment. For the “Grace” sequence, to achieve 40% average error, the IsoSW approach needs 140 important coefficients. When the CSW approach is used, we need to use 180 important coefficients. For the “ST. PETER” sequence, to achieve 40% average error, the IsoSW approach needs 160 important coefficients. When the CSW approach is used, we need to use more than 200 important coefficients.

The above experimental results confirm that the IsoSW approach has a better representation ability. This is because the IsoSW approach has the uniform sampling pattern. It provides the same frequency analysis on the important features no matter where the important features are. On the other hand, the CSW approach has more sample points on the face corner regions. Hence, if important features are on these regions, we need more wavelet coefficients to represent the important feature.

B. Error in Synthesized Results

In the second experiment, we evaluate the rendering quality. For each HDR environment sequence, we render images of the testing 3D scenes by taking each environment frame as the lighting environment separately. Suppose $I(t)$ is the rendered control result for the t-th environment frame, and $\hat{I}(t)$ is the rendered image by using a set of important wavelet coefficients. Then, the average percentage error of rendered images ($E_I(t)$), is computed as,

$$\tilde{E}_I = \frac{1}{T}\sum_t E_I(t) = \frac{1}{T}\sum_t \frac{\sum_{x,y}(\hat{I}(x,y,t) - I(x,y,t))^2}{\sum_{x,y}(I(x,y,t))^2}. \qquad (16)$$

The graphs in Figure 14 compare the average percentage error of rendered images for different experimental settings, including TEAPOT under GRACE, TEAPOT under ST. PETER, BUNNY under GRACE and BUNNY under ST. PETER. Similar to the results of the lighting approximation in Figure 13, the more the coefficients are used, the less the average percentage error will be. Compared with the CSW approach, the IsoSW approach achieves smaller percentage errors for a given number of wavelet coefficients.

The determination of number of important coefficients depends on the environment lighting complexity and the reflectance of 3D scene. Ng's paper [12] mentioned that around 100 to 200 coefficients resolve essentially complex shadow details [12]. As shown in Figure 14, when 100 coefficients are used, the average percentage error of the CSW approach for (a) TEAPOT under GRACE, (b) TEAPOT under ST. PETER, (c) BUNNY under GRACE and (d) BUNNY under ST. PETER are 0.066%, 0.028%, 0.475% and 0.272%, respectively. For the same percentage errors, the IsoSW approach only needs 70, 80, 75 and 75 coefficients, respectively.

When 200 coefficients are used, the average percentage error of the CSW approach for (a) TEAPOT under GRACE, (b) TEAPOT under ST. PETER, (c) BUNNY under GRACE and (d) BUNNY under ST. PETER are 0.0207%, 0.00768%, 0.126% and 0.0638%, respectively. For the same percentage errors, the IsoSW approach only needs 150, 180, 190 and 180 coefficients, respectively. Clearly, the IsoSW approach is more effective in the lighting approximation.

Next, we explore how the two approaches perform over time given the same amount of important wavelet coefficients. In Figure 15, we fix the number of important wavelet coefficients to be 100, and compare the percentage errors ($E_I(t)$) of rendered images for the four experimental settings. In the four cases, the IsoSW (in blue color) achieves smaller errors for most rendered frames. Also, we observe that there are large variations in percentage errors in the CSW case (in red color) over time. In other words, the IsoSW approach can outperform the CSW approach in many cases.

Careful readers may notice that the percentage error of the rendered results are smaller than that of the lighting environment. This is because the rendered results are affected not only by the lighting, but by the geometry and the reflectance properties of the underlying 3D scenes as well. Hence, although the difference of the errors for the lighting approximation between the CWS approach and the IsoSW approach is large, the difference of the errors for the synthesized images usually is small.

C. Visual Comparison

We now explore whether the synthesized images have differences in visual quality as the average percentage errors indicate. Figure 16 and 17 visually compare the synthesized images from the CSW and IsoSW approaches. They are the rendered images of TEAPOT under GRACE and BUNNY under ST. PETER, respectively. In these two experiments, the dense sampling point set 81,920 is used to generate the control images.

In the TEAPOT under GRACE case, we select three different numbers of important coefficients, i.e., 49, 100 and 200 among 6,144 coefficients. The percentage errors are listed under the rendered results. When using frame 5 in GRACE sequence as the lighting source, both approaches provide similar visual quality for all the tested number of coefficients. Compared with the control result, the highlight on the teapot body is more blurry and scattered when the number of coefficients is only 49, while it becomes more focused when the number of coefficients increases. A closer inspection reveals the unexpected abrupt changes in the highlight of teapot body when the CSW approach is used for synthesis, although they become less visible when more coefficients are selected. In contrast, no such artifacts are observed in our IsoSW results. As the visual differences may be less distinctive when printed out, the images are enhanced and readers are referred to the PDF file for a clearer comparison.

In the BUNNY under ST. PETER case, both approaches provide similar visual quality when 200 coefficients are used. However, when we use 60 coefficients only, there are serious banding artifacts around the shadows on the ground for the CSW approach. Even when 100 coefficients are selected, the artifacts for the CSW approach are still apparent. On the other hand, the IsoSW approach reaches a reasonable visual quality when using 100 coefficients. In other words, the IsoSW approach can improve the rendering efficiency by using fewer coefficients.

To further discuss the under-sampling artifacts, we try to find out an environment frame which contains high-frequency illumination at the center of a face. We select frame 42 of GRACE sequence as the lighting environment (the second frame in Figure 12(a)). The rendered images of TEAPOT are shown in Figure 18. In this example, we find that even when a relatively large number of coefficients are used, the rendering results generated using the CSW approach are still poor. The artifacts in the highlight on the TEAPOT body are significantly observable for all the tested numbers of coefficients. These artifacts are generated due to the uneven-sampling of the cubemap faces. To suppress the uneven-sampling artifacts, we have to use more dense samples and more wavelet coefficients for the CSW approach. The results from our IsoSW approach, on the other hand, are much better. Note that no additional cost is needed for the suppression of under-sampling artifacts in the IsoSW approach.

V. Conclusion

In this paper, we present a new approach, IsoSW approach, to facilitate the rendering of IAI under time-varying environment lighting. By casting a highly uniform sampling pattern, Isocube mapping, over the spherical domain, our IsoSW approach can efficiently extract variations in both the IAI and the environment lighting, even though no prior information is available about the data distribution. Consequently, our approach not only suppresses the under-sampling artifacts, but also preserves high-frequency features, such as shadows and highlights, in a decent rendering condition. As evidenced in our experiments, we can achieve better rendering quality while using fewer number of wavelet coefficients as compared to the CSW approach. Therefore, our approach outperforms the previous approach by providing a more balanced rendering quality and better efficiency in both computation and storage.

ACKNOWLEDGMENT

The work was supported by a research grant from City University of Hong Kong (Project No.: 7002608).

REFERENCES

[1] Wong, T.T., Heng, P.A., Or, S.H., and Ng, W.Y.: 'Image-based rendering with controllable illumination', Proc. 8th Eurographics Workshop on Rendering, Saint Etienne, France, June 1997, pp. 13–22

[2] Kautz, J., Sloan, P.P., and Snyder, J.: 'Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics', Proc. 13th Eurographics Workshop on Rendering, Pisa, Italy, June 2002, pp. 291–297

[3] Debevec, P.: 'Image-based lighting', *IEEE Comput. Graphics Appl.*, 2002, 22, (2), pp. 26–34

[4] Veach, E., and Guibas, L.J.: 'Optimally combining sampling techniques for monte carlo rendering', SIGGRAPH95, Los Angeles, California, USA, August 1995, pp. 419–428

[5] Zongker, D.E., Werner, D.M., Curless, B., and Salesin, D.H.: 'Environment matting and compositing', SIGGRAPH99, Los Angeles, California, USA, August 1999, pp. 205–214

[6] Debevec, P., Hawkins, T., Tchou, C., Duiker, H.P., Sarokin, W., and Sagar, M.: 'Acquiring the reflectance field of a human face', SIGGRAPH00, New Orleans, Louisiana, USA, July 2000, pp. 145–156

[7] Wong, T.T., Fu, C.W., and Heng, P.A.: 'Interactive relighting of panoramas', *IEEE Comput. Graphics Appl.*, 2001, 21, (2), pp. 32–41

[8] Lin, Z., Wong, T.T., and Shum, H.Y.: 'Relighting with the reflected irradiance field: Representation, sampling and reconstruction', *Int. J. Comput. Vision*, 2002, 49, (2-3), pp. 229–246

[9] Wong, T.T., Fu, C.W., Heng, P.A., and Leung, C.S.: 'The plenoptic illumination function', *IEEE Trans. Multimedia*, 2002, 4, (3), pp. 361–371

[10] Wong, T.T., and Leung, C.S.: 'Compression of illumination adjustable image', *IEEE Trans. Circuits Syst. Video Technol.*, 2003, 13, (11), pp. 1107–1118

[11] Macrobert, T.: 'Spherical harmonics: an elementary treatise on harmonic functions with applications' (Dover Publications, 1948)

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

[13] Schröder, P., and Sweldens, W.: ‘Spherical wavelets: efficiently representing functions on the sphere’, SIGGRAPH95, Los Angeles, California, USA, August 1995, pp. 161–172

[14] Wang, Z., Leung, C.S., Zhu, Y.S., and Wong, T.T.: ‘Data compression with spherical wavelets and wavelets for the image-based relighting’, *Comput. Vision Image Understanding*, 2004, 96, (3), pp. 327–344

[15] Gorski, K.M., Wandelt, B.D., Hivon, E., Hansen, F.K. and Banday, A.J.: ‘The HEALPix primer’, Techical Report, Theoretical Astrophysics Center (TAC) Kopenhagen, feb 2003, astro-ph/9905275.

[16] Wan, L., Wong, T.T., and Leung, C.S.: ‘Isocube: Exploiting the cubemap hardware’, *IEEE Trans. Visual Comput. Graphics*, 2007, 13, (4), pp. 720–731

[17] Debevec, P., and Malik, J.: ‘Recovering high dynamic range radiance maps from photographs’, SIGGRAPH97, August 1997, pp. 369–378

[18] Liu, X., Sloan, P.P., Shum, H.Y., and Snydeer, J.: ‘All-frequency precomputed radiance transfer for glossy objects’, Proc. 15th Eurographics Symposium on Rendering, Norrk*ö*ping, Sweden, 2004, pp. 337–344

[19] MALLAT, S.: ‘A Wavelet Tour of Signal Processing’ (Academic Press, 1999)

[20] Greene, N.: ‘Environment mapping and other applications of world projections’, *IEEE Computer Graphics and Applications*, 1986, 6, (11), pp. 21–29

Figure legends:

- Figure 1

  TEAPOT illuminated by an area light source from above. (a) rendering result using our isocube-based spherical wavelet approximation. (b) the blowup of the control image using 81,920 directional lights that uniformly sample the environment lighting. (c) the blowup from the cubemap-based spherical wavelets. (d) the blowup from the isocube-based spherical wavelets. Both (c) and (d) are generated using only 200 most important wavelet coefficients.

- Figure 2

  The view–plane organization in the PIF model.

- Figure 3

  Sampling the lighting direction on the isocube.

- Figure 4

  Cube mapping (a) a cube with regular grid lines on each of cube faces. (b) The unrolled map of the cube. (c) The partitioning of a sphere based on cube mapping. (d) Sampling patterns on the sphere.

- Figure 5

  Sampling a spherical function with the cube mapping. (a) a spherical function. (b) the sampled spherical functions shown in in the unrolled view of the cube map. The resolution of the cubemap is $6 \times 16 \times 16$.

- Figure 6

  Visualization of CSW basis functions. For a cubemap with resolution of $6 \times 4 \times 4$, there are 96 basis functions. In this figure, we show 32 of them. In the figure, positive values are represented by gray colors, while negative values are represented by red colors.

- Figure 7

  Isocube partitioning: (a) equatorial and polar zones in 3D views. (b) the 6 equal-area base faces produced from the first-level partitioning.

- Figure 8

  Recursive subdivision of isocube at different resolutions.

- Figure 9

  Sampling a spherical function with Isocube mapping (a) a spherical function. (b) the sampled spherical functions shown in in the unrolled view of the Isocube map. The resolution of the Isocube map is $6 \times 16 \times 16$. fig:samplingiso

- Figure 10

  Grouping the radiance values related to the same pixel.

- Figure 11

  The construction of isoSW lightmaps.

- Figure 12

  Some environment frames: (a) in the sequence GRACE; and (b) in the sequence ST. PETER.

- Figure 13

  Average percentage errors of lighting approximation: (a) GRACE and (b) ST. PETER.

- Figure 14

  Average percentage errors of synthesized images with environment frame sequences: (a) TEAPOT under GRACE, (b) TEAPOT under ST. PETER, (c) BUNNY under GRACE and (d) BUNNY under ST. PETER.

- Figure 15

  Percentage errors of synthesized images with environment frame sequences using 100 coefficients: (a) TEAPOT under GRACE, (b) TEAPOT under ST. PETER, (c) BUNNY under GRACE and (d) BUNNY under ST. PETER.

- Figure 16

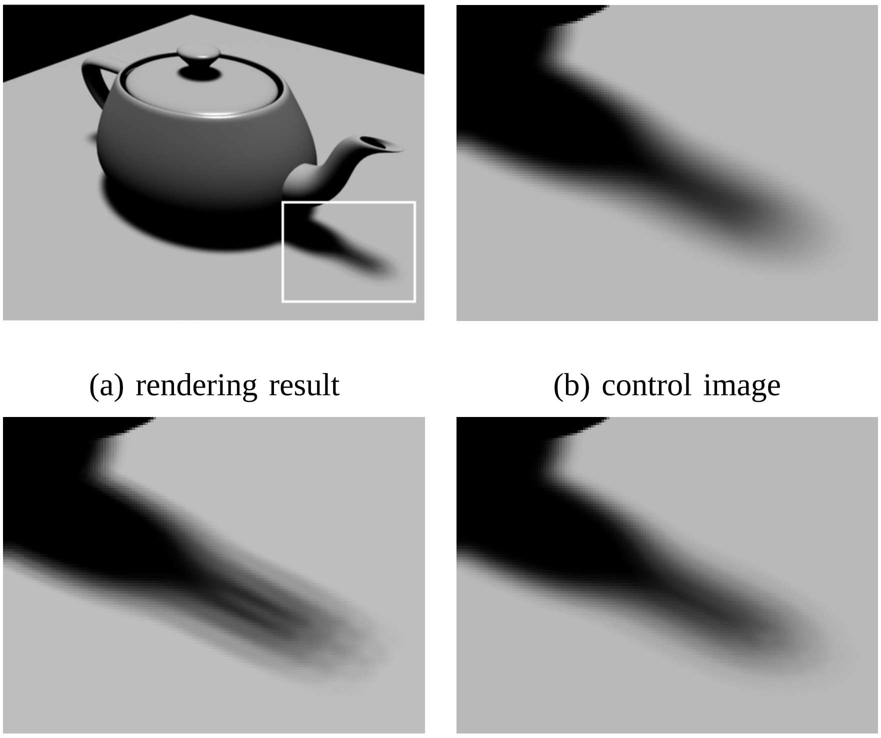
  Rendered images of TEAPOT. The lighting environment in use is frame 5 in GRACE sequence. Note the artifacts in the highlight of TEAPOT body when the CSW approach is used for rendering. In contrast, our isoSW approach has no such artifacts. [*As the visual differences may be less distinctive when printed out, readers are referred to the PDF file for a clearer comparison.*]

- Figure 17

  Rendered images of BUNNY. The lighting environment in use is frame 17 in ST. PETER. Note the visual artifacts in the shadows on the ground when using the CSW approach with 60 coefficients. The artifacts are still apparent when using the CSW approach with 100 coefficients.

- Figure 18

  TEAPOT rendered using frame 42 of GRACE sequence. Note that in frame 42 of GRACE, there is a strong area light source located at a cubemap face center. Hence, even we use as many as 200 wavelet coefficients in the rendering, there are still under-sampling artifacts on the TEAPOT body. In our isoSW approach, the artifacts are much reduced.



(a) rendering result

(b) control image

(c) CSW

(d) isoSW

Fig. 1. TEAPOT illuminated by an area light source from above. (a) rendering result using our isocube-based spherical wavelet approximation. (b) the blowup of the control image using 81,920 directional lights that uniformly sample the environment lighting. (c) the blowup from the cubemap-based spherical wavelets. (d) the blowup from the isocube-based spherical wavelets. Both (c) and (d) are generated using only 200 most important wavelet coefficients.

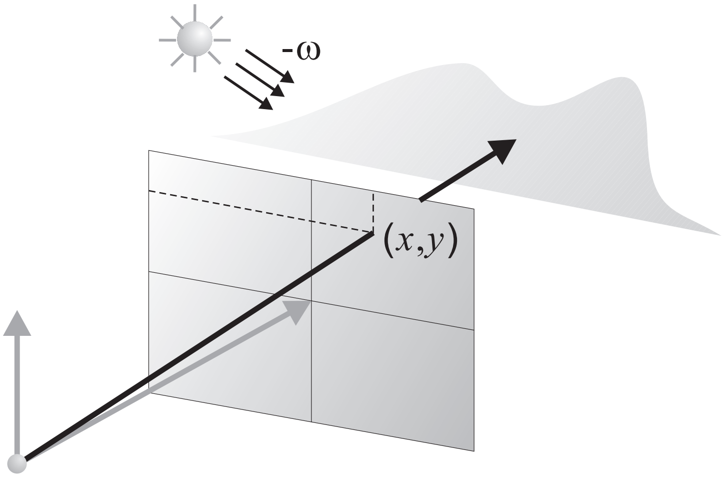


Fig. 2. The view–plane organization in the PIF model.

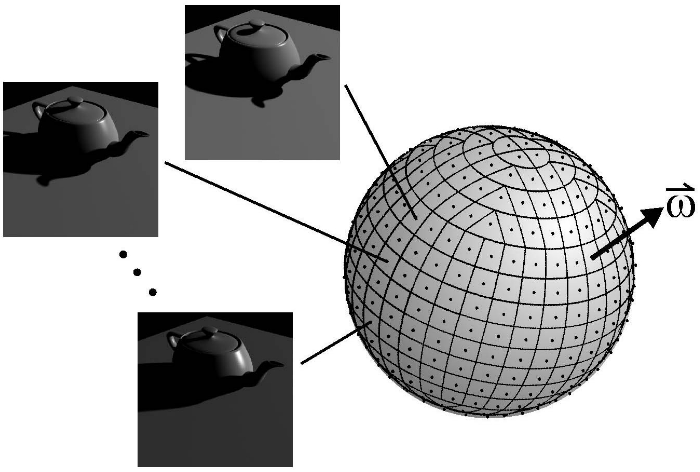


Fig. 3. Sampling the lighting direction on the isocube.

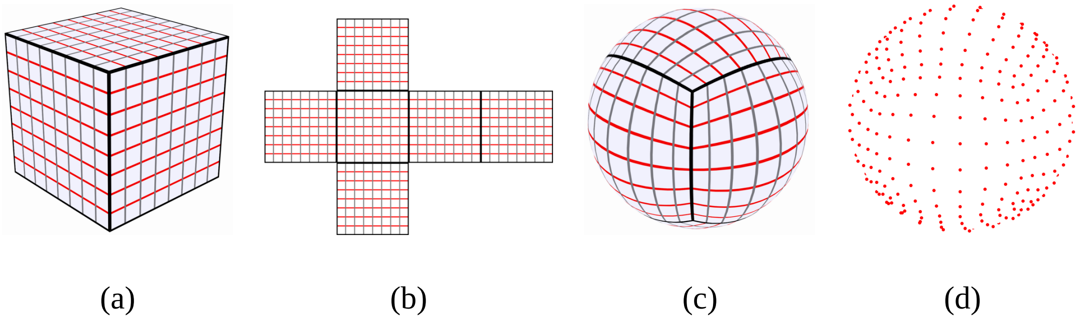


Fig. 4. Cube mapping (a) a cube with regular grid lines on each of cube faces. (b) The unrolled map of the cube. (c) The partitioning of a sphere based on cube mapping. (d) Sampling patterns on the sphere.

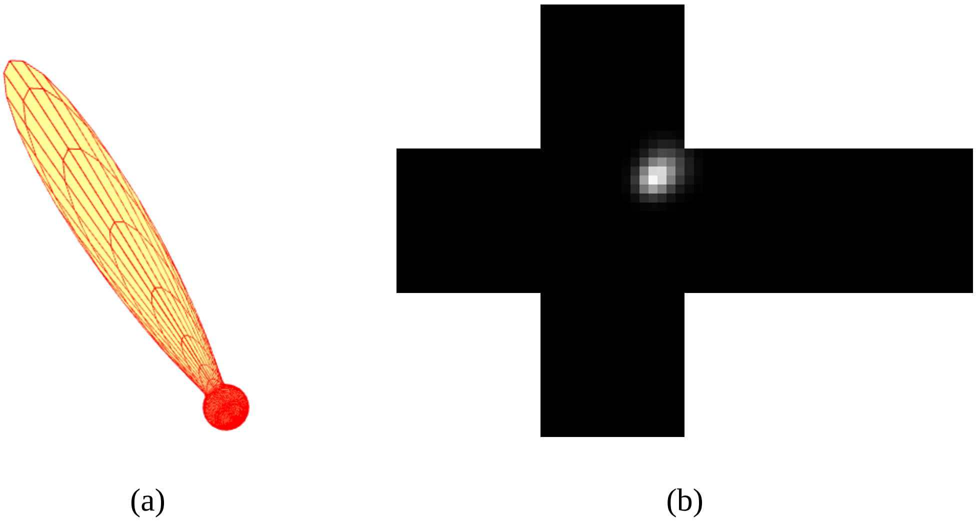


Fig. 5. Sampling a spherical function with the cube mapping. (a) a spherical function. (b) the sampled spherical functions shown in in the unrolled view of the cube map. The resolution of the cubemap is $6 \times 16 \times 16$.

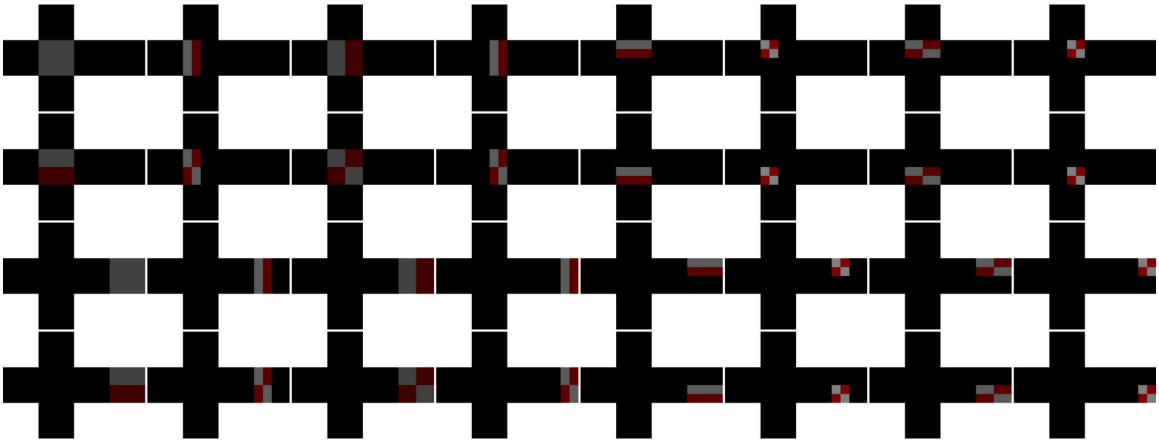


Fig. 6. Visualization of CSW basis functions. For a cubemap with resolution of $6 \times 4 \times 4$, there are 96 basis functions. In this figure, we show 32 of them. In the figure, positive values are represented by gray colors, while negative values are represented by red colors.

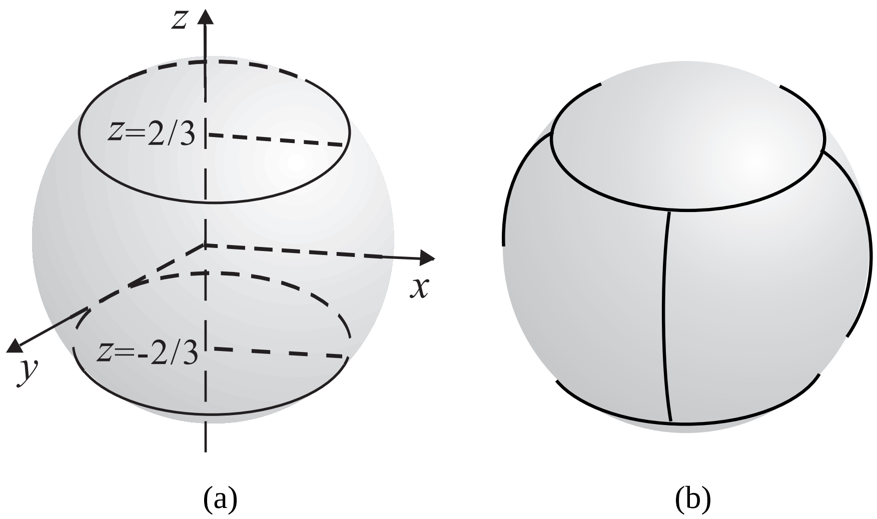
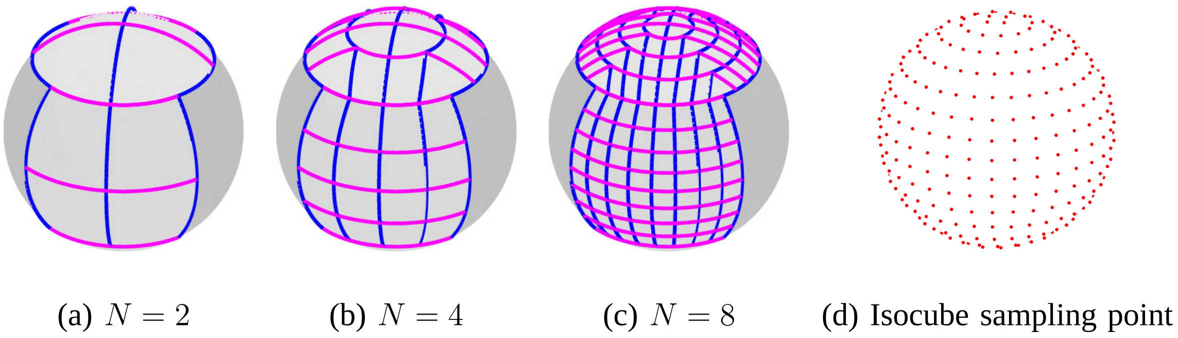
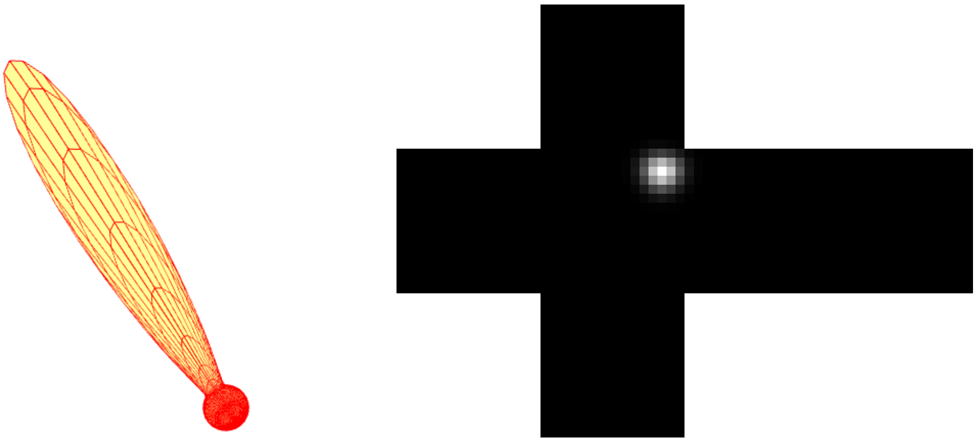


Fig. 7. Isocube partitioning: (a) equatorial and polar zones in 3D views. (b) the 6 equal-area base faces produced from the first-level partitioning.



(a) $N = 2$ (b) $N = 4$ (c) $N = 8$ (d) Isocube sampling point

Fig. 8. Recursive subdivision of isocube at different resolutions.



(a)

(b)

Fig. 9. Sampling a spherical function with Isocube mapping (a) a spherical function. (b) the sampled spherical functions shown in in the unrolled view of the Isocube map. The resolution of the Isocube map is $6 \times 16 \times 16$.

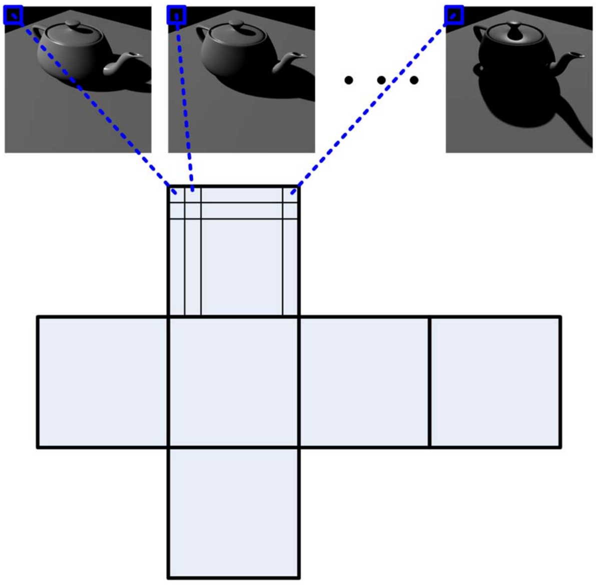


Fig. 10. Grouping the radiance values related to the same pixel.

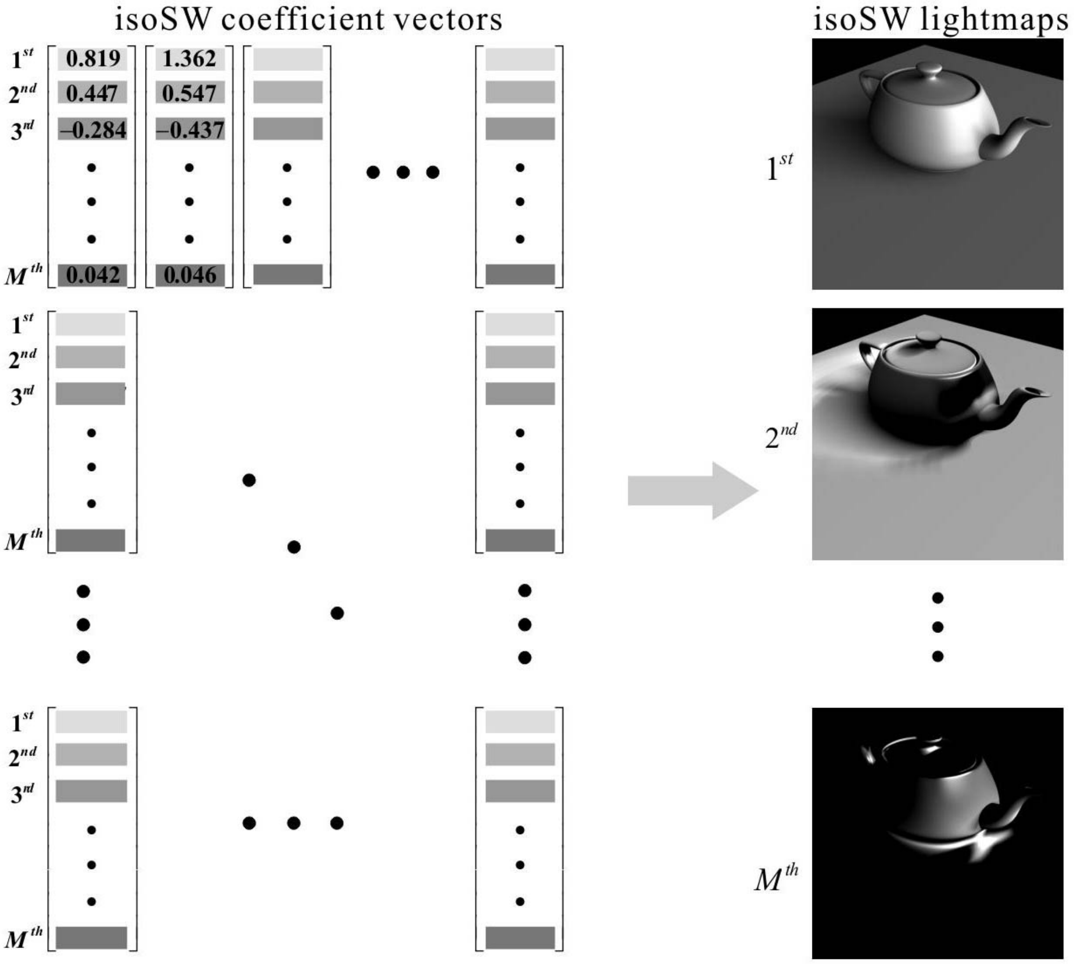


Fig. 11. The construction of isoSW lightmaps.

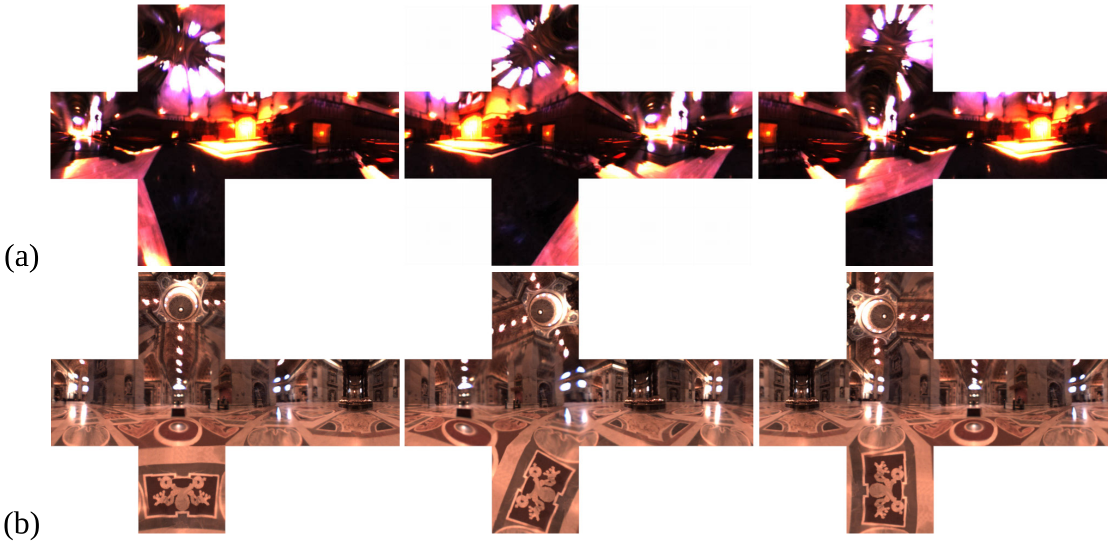
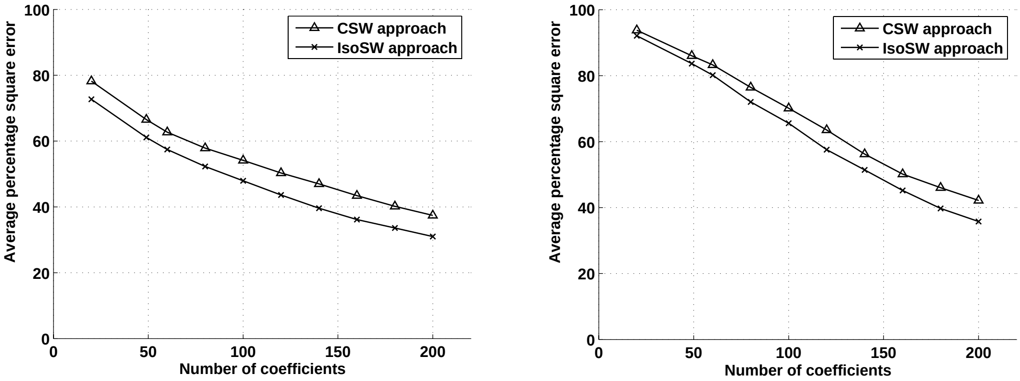


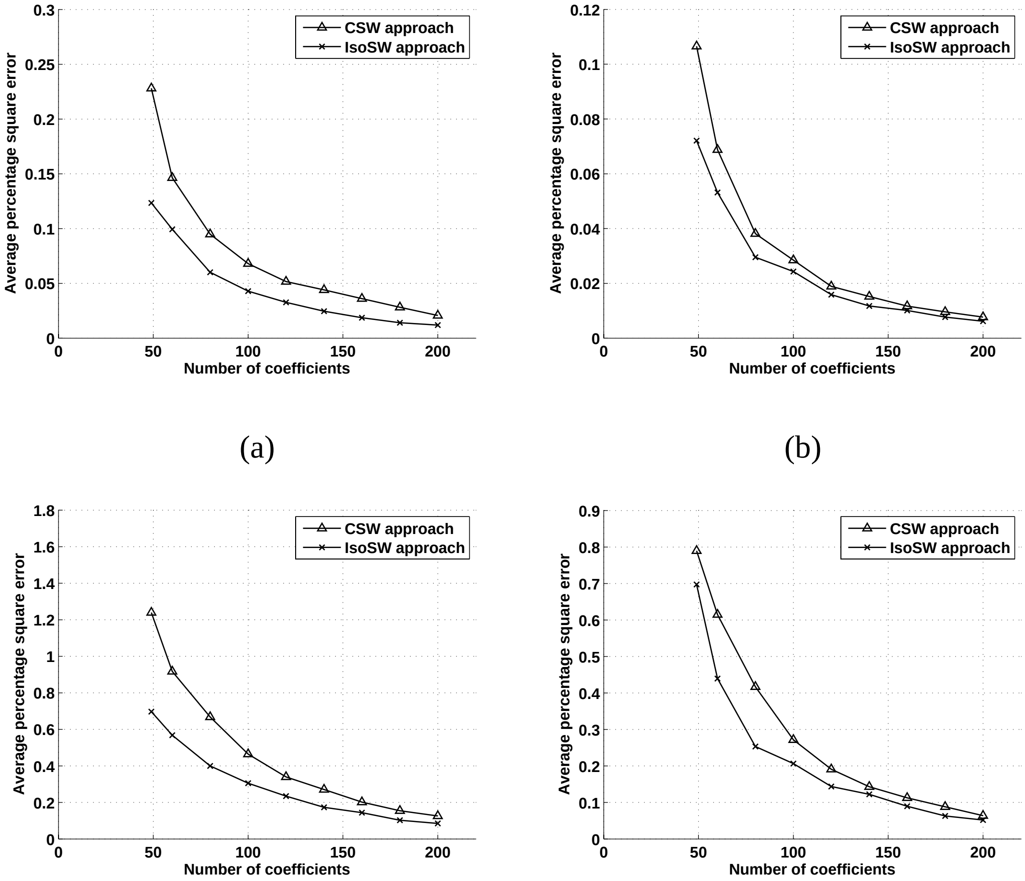
Fig. 12. Some environment frames: (a) in the sequence GRACE; and (b) in the sequence ST. PETER.



(a) GRACE

(b) ST. PETER

Fig. 13. Average percentage errors of lighting approximation: (a) GRACE and (b) ST. PETER.



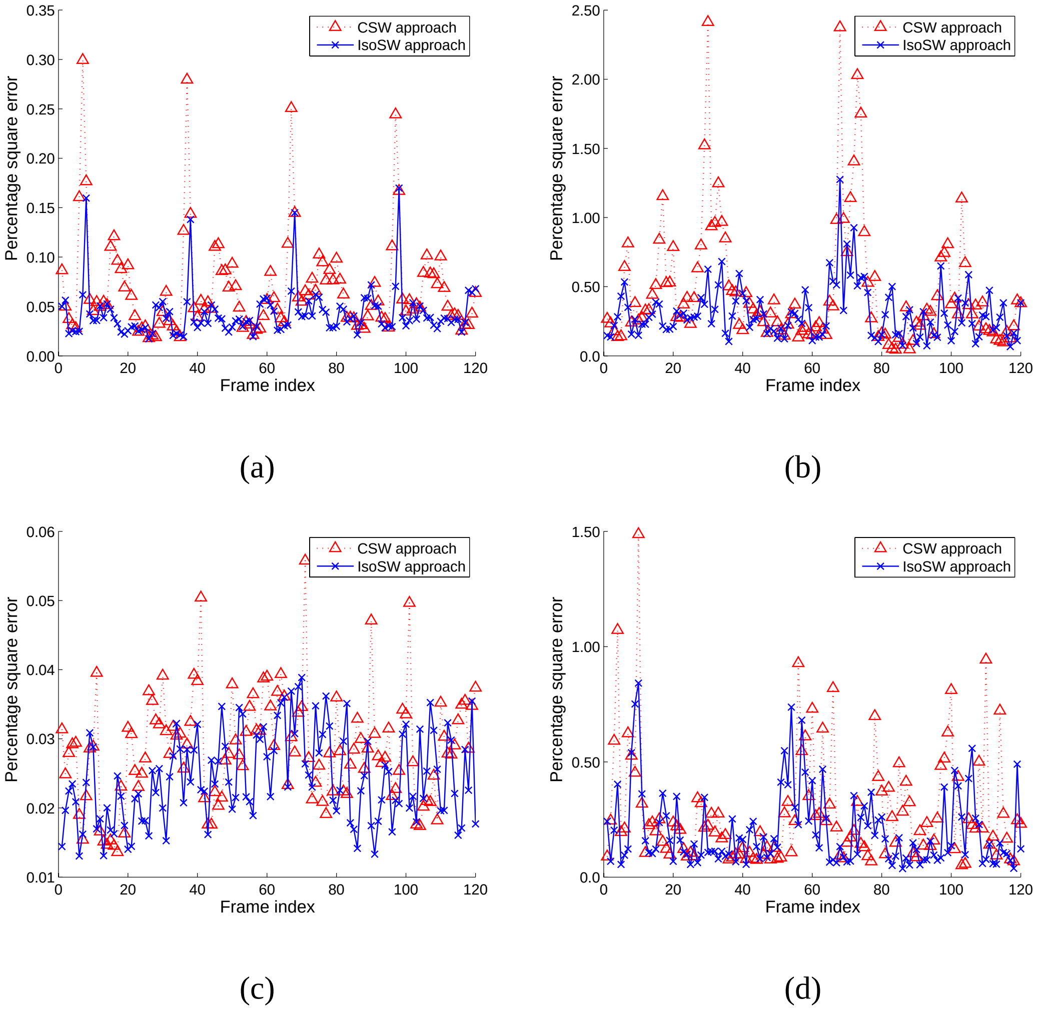
(a)

(b)

(c)

(d)

Fig. 14. Average percentage errors of synthesized images with environment frame sequences: (a) TEAPOT under GRACE, (b) TEAPOT under ST. PETER, (c) BUNNY under GRACE and (d) BUNNY under ST. PETER.



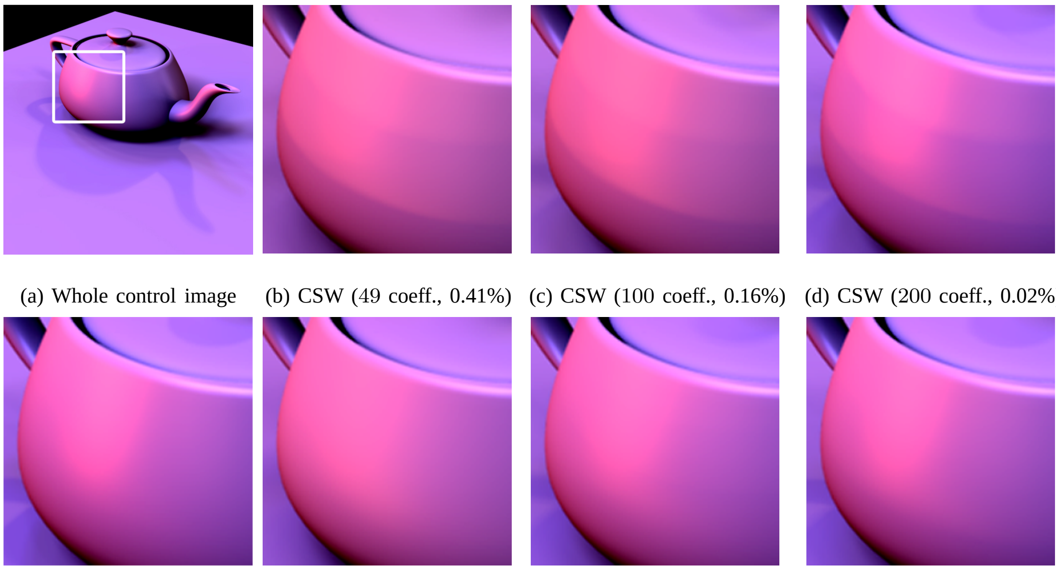
(a)

(b)

(c)

(d)

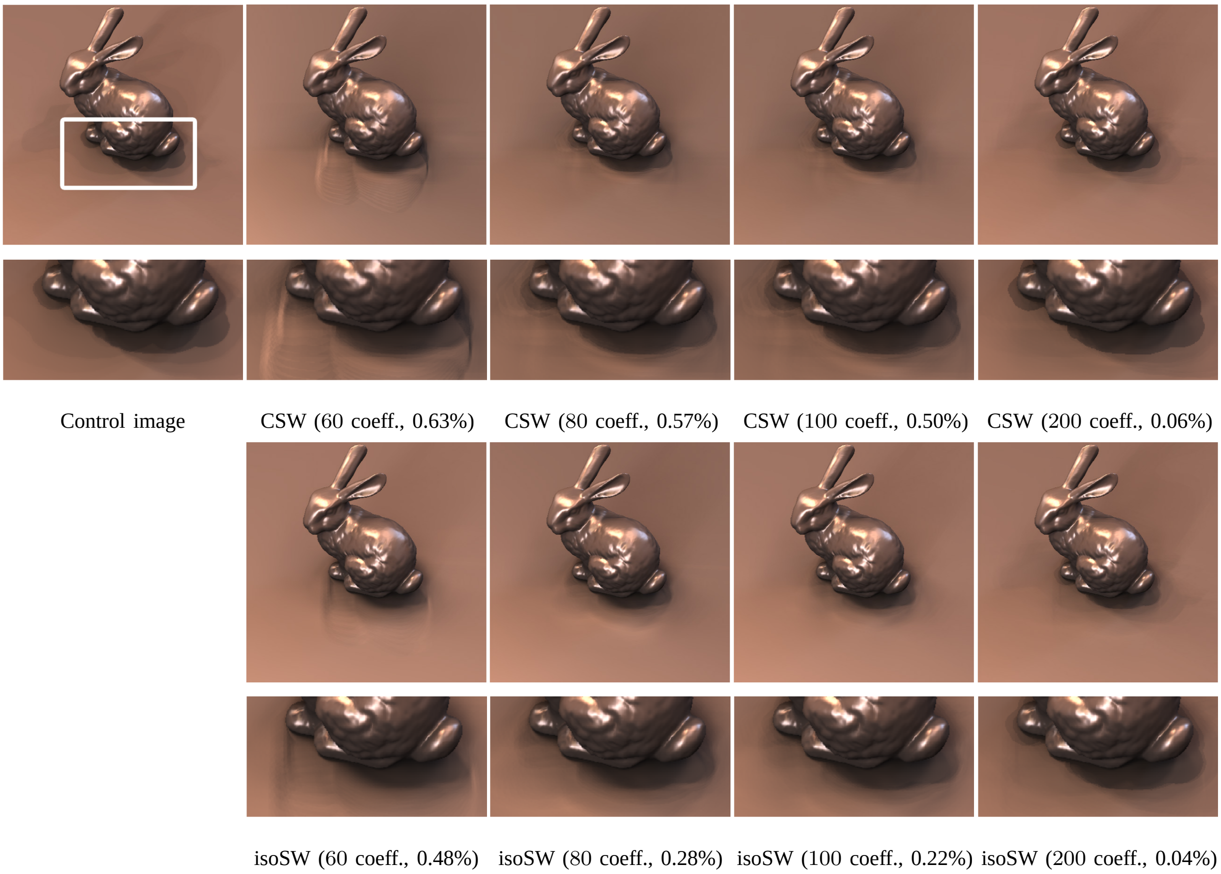
Fig. 15. Percentage errors of synthesized images with environment frame sequences using 100 coefficients: (a) TEAPOT under GRACE, (b) TEAPOT under ST. PETER, (c) BUNNY under GRACE and (d) BUNNY under ST. PETER.



(a) Whole control image (b) CSW (49 coeff., 0.41%) (c) CSW (100 coeff., 0.16%) (d) CSW (200 coeff., 0.02%)

(e) Control image (f) isoSW (49 coeff., 0.14%) (g) isoSW (100 coeff., 0.02%) (h) isoSW (200 coeff., 0.01%)

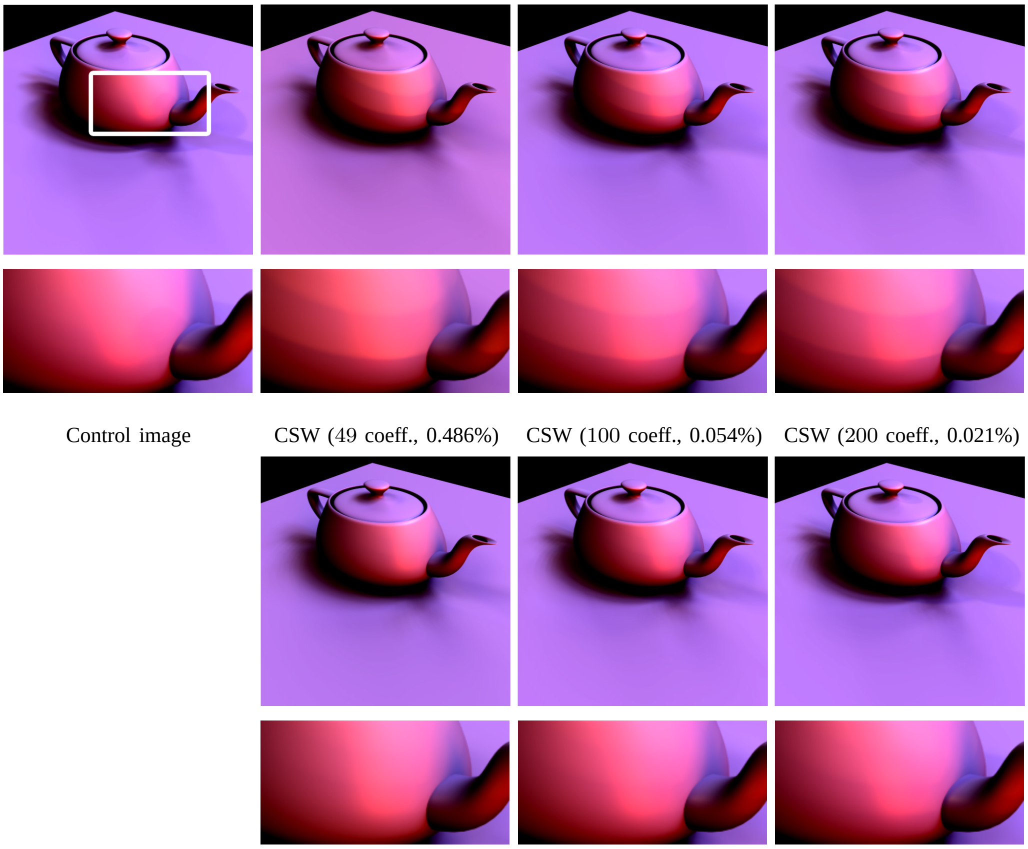
Fig. 16. Rendered images of TEAPOT. The lighting environment in use is frame 5 in GRACE sequence. Note the artifacts in the highlight of TEAPOT body when the CSW approach is used for rendering. In contrast, our isoSW approach has no such artifacts. [*As the visual differences may be less distinctive when printed out, readers are referred to the PDF file for a clearer comparison.*]



Control image CSW (60 coeff., 0.63%) CSW (80 coeff., 0.57%) CSW (100 coeff., 0.50%) CSW (200 coeff., 0.06%)

isoSW (60 coeff., 0.48%) isoSW (80 coeff., 0.28%) isoSW (100 coeff., 0.22%) isoSW (200 coeff., 0.04%)

Fig. 17. Rendered images of BUNNY. The lighting environment in use is frame 17 in ST. PETER. Note the visual artifacts in the shadows on the ground when using the CSW approach with 60 coefficients. The artifacts are still apparent when using the CSW approach with 100 coefficients.



Control image CSW (49 coeff., 0.486%) CSW (100 coeff., 0.054%) CSW (200 coeff., 0.021%)

isoSW (49 coeff., 0.139%) isoSW (100 coeff., 0.033%) isoSW (200 coeff., 0.007%)

Fig. 18. TEAPOT rendered using frame 42 of GRACE sequence. Note that in frame 42 of GRACE, there is a strong area light source located at a cubemap face center. Hence, even we use as many as 200 wavelet coefficients in the rendering, there are still under-sampling artifacts on the TEAPOT body. In our isoSW approach, the artifacts are much reduced.