

Compression of Illumination-Adjustable Images

Tien-Tsin Wong^{*1} Chi-Sing Leung²

ttwong@acm.org

eeleungc@cityu.edu.hk

¹ Dept. of Computer Science & Eng., The Chinese University of Hong Kong

² Dept. of Electronic Eng., City University of Hong Kong

Tien-Tsin Wong (* Contact author)

Dept. of Computer Science & Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong.

Tel: +852-26098433 Fax: +852-26035024 Email: ttwong@acm.org

Chi-Sing Leung

Dept. of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong.

Tel: +852-27887378 Fax: +852-27887791 Email: eeleungc@cityu.edu.hk

This research was supported by the RGC Direct Grant for Research, the Chinese University of Hong Kong (Project ID: 2050262) and the Strategic Grant, City University of Hong Kong (Project No. 7001079).

Abstract

The image-based modeling and rendering (IBMR) approaches allow the time complexity of synthesizing novel images to be independent of scene complexity. Unfortunately, illumination control (re-lighting) is no longer trivial under the image-based framework. To relight the image-based scenery, the scene must be captured under various illumination conditions. This drastically increases the data volume. Hence, data compression is a must. In this paper, we describe a compression algorithm for an illumination-adjustable image representation. We focus on compressing constant-viewpoint images. A divide-and-conquer approach is proposed. The compression algorithm consists of three parts which exploit the *intra-pixel*, *inter-pixel* and *inter-channel* data correlations. Experimental result shows that the proposed method not just effectively compresses the data but also out-performs standard image and video coding methods.

Keywords

Image-based modeling and rendering, image-based relighting, multimedia application, data compression, spherical harmonic transform, wavelet transform.

I. INTRODUCTION

Image-based modeling and rendering (IBMR) is an alternative to the traditional geometry-based computer graphics. The advantage is that its rendering time complexity can be decoupled from the unbounded scene complexity. Hence, interactive or even real-time rendering of arbitrary complex scene becomes possible. The commercial application, QuickTime VR [1], demonstrates the efficiency and practicability of the IBMR approach. However, since the introduction of QuickTime VR, not many IBMR techniques have been successfully applied to Internet applications. One possible reason is its tremendous storage requirement as the IBMR approaches actually trade storage for speed.

Besides the storage issue, functions that are trivial in the traditional geometry-based approach become difficult under the image-based framework. For examples, moving the viewpoint and changing the lighting condition both become difficult. Most previous IBMR researches focus on view interpolation. Less attention has been paid on *image-based relighting* (the ability of illumination control). We have previously proposed an illumination-adjustable image representation [2], [3], apparent pixel BRDF (Bidirectional Reflectance Distribution Function), which allows illumination to be adjusted under the IBMR framework. With this representation, we can *relit* panoramas with novel lighting configurations in an interactive [4] or even a real-time [5]

manner. Hence, our illumination-adjustable image representation can be applied in computer game and indoor lighting design which require real-time modification of lighting condition. The potentiality is demonstrated in our previously developed illumination-adjustable panorama viewer [5]. However, just like other IBMR techniques, the major drawback of this representation is the enormous storage requirement.

In this paper, we propose a compression method for this illumination-adjustable image representation. In this representation, a pixel is not only associated with a single radiance value but a number of radiance values captured under various illumination conditions. We consider the compression of radiance values captured from a constant viewpoint. These *constant-viewpoint illumination-adjustable images* can be planar perspective, panoramic or even fish-eye. Since we handle radiance data captured under various illumination conditions, the constant-viewpoint image is a set of thousands of reference images. For a $1,024 \times 256$ illumination-adjustable image captured under 1,200 illumination conditions (1,200 is the empirical number of samples), the data size without any compression is 900MB. Even with highly compressed (low image quality) JPEG encoding, the essential data size is still around 100MB, which is not practical for Internet applications.

Our proposed method out-performs JPEG, JPEG2000 and MPEG because these three standard methods cannot fully exploit the correlation in reference images. In our method, we divide the complex compression problem into three main steps which exploit three kinds of data correlations, namely *intra-pixel*, *inter-pixel* and *inter-channel* correlations.

The intra-pixel correlation refers to the correlation of radiance values of a pixel captured under various illumination conditions. As the viewing direction is constant for all images, these radiance values are closely related to (but not equal to) the reflectance of surface element visible through the pixel window. We transform the data to a frequency domain using the spherical harmonic transform. The quantization is then performed in the spherical harmonic domain to reduce the data size drastically. Section IV describes this first-step compression in details. Note that this correlation can be regarded as *inter-image* correlation in the traditional video coding approaches because we group values from different images. Since our method stems from the concept of apparent pixel BRDF [2], it is more natural to use the “pixel” terminologies.

The inter-pixel correlation refers to the correlation of data between adjacent pixels. This is

commonly referred as *intra-image* correlation and exploited in the traditional image coding approaches. Note that, after intra-pixel compression, each pixel no longer contains the radiances, but a vector of spherical harmonic coefficients. In Section V, we show that the inter-pixel correlation also exists among these coefficient vectors even they are in the spherical harmonic domain. Hence, a wavelet compression technique is applied.

Finally, we exploit the inter-channel correlation in Section VI as we handle color images. Human perception is not equally sensitive to every color channel. Data reduction can be achieved by transforming the RGB model to other color model and reducing bandwidths of those less-sensitive channels. We shall discuss how such correlation can be exploited in spherical harmonic domain.

This paper is organized in the following structure. Related works are reported in Section II. For completeness, we then describe in Section III the illumination-adjustable image representation we have previously proposed. Sections IV, V and VI describe the three main compression steps. An overall performance evaluation is presented in Section VII. Finally, conclusions are drawn and future directions are discussed in Section VIII.

II. RELATED WORK

Images have long been used in computer graphics for approximating the microscopic surface details. This application is commonly known as texture mapping [6]. In the recent work, images are used for approximating complex scenery. Several IBMR approaches [7], [8] have been proposed. A stream of IBMR techniques [8], [3] treats the problem as a sampling and reconstruction of the ideal plenoptic function [9]. Its beauty is simplicity, but with a drawback of enormous storage requirement. There are several attempts to reduce the storage by reducing dimensionality, determining the minimum sampling bound, and removing redundancy.

A. Dimensionality Reduction

As mentioned in [8], the IBMR approach can be treated as a problem of sampling and reconstruction of the seven-dimensional *plenoptic function* [9]. Levoy and Hanrahan [10] and Gortler *et al.* [11] reduced the plenoptic function to a 4D light field or Lumigraph. Two parallel planes are used for parameterizing any ray passing through the enclosed volume. Shum and He [12] further reduced its dimension to 3D by restricting the viewpoint on concentric circles. Chen [1]

generated perspective views by warping the cylindrical panoramic image (2D plenoptic function) in his QuickTime VR system.

Besides view interpolation, another important goal of IBMR is *relighting*. By saying relighting, we mean synthesizing images with novel lighting conditions given a set of reference images. However, the dimensionality of the plenoptic function has to be increased by two when the illumination is included [3].

B. Sampling Sufficiency

As a novel image is synthesized by interpolating reference images captured, the quality of the synthesized images depends on the number of reference images. Chai *et al.* [13] recently showed that the minimum-sampling curve for light field or Lumigraph can be determined given the rough depth information of the scene. With this curve, unnecessary samples can be avoided which in turn reduces the storage requirement.

On the other hand, Lin *et al.* [14] derived the theoretical sampling bound for image-based relighting. However, even with the guidance of the sampling bound, the storage requirement is still enormous. This motivates our work presented in this paper.

C. Redundancy Removal

Several compression techniques have been proposed to remove data redundancy in the image-based data. Vector quantization and wavelet transform have been used for compressing light field [15] [16] [17]. To reduce the data volume of concentric moasics, Zhang and Li [18] proposed a reference block codec.

To remove the redundancy in image-based relighting data, the principal component analysis [19] can be used. The basic idea is to determine a minimal set of basis images (eigen-images) from the given reference images. Several variants [20] [21] [22] have been proposed recently. The desired images can be synthesized by linearly combining these eigen-images given a set of eigen-coefficients. Unfortunately, eigen-coefficients cannot be related to the lighting direction by a simple mathematical equation. Nishino *et al.* [23] related eigen-coefficients to the light vector by a table-lookup method. In their system, the light source can be controlled in a circular trajectory. Although empirical experiments [22] show that 3 to 7 eigen-images may be good enough to represent specular surfaces, the quality of the relit images also depends on the scene

geometry. Bumpy surfaces may produce much highlight which is hard to be represented by a few eigen-images. Specular highlight and shadow are still difficult to be accurately represented.

Transform coding is another effective tool to remove redundancy in relighting data. Wong *et al.* [2] treated each pixel in the image as an abstract surface element and recorded its apparent BRDF. Desired images can be synthesized by looking up the recorded apparent BRDFs. Spherical harmonic coefficients are used to efficiently represent the BRDFs. For diffuse reflection, Ramamoorthi and Hanrahan [24] proved that 9 spherical harmonic coefficients are sufficient. They also applied the spherical harmonic transform to represent the sophisticated illumination model [25] [26] [27].

Malzbender *et al.* [28] approximated the apparent pixel BRDF by fitting a polynomial function. Since only 6 parameters are needed for each pixel, a high compression ratio is achieved. However, due to the limited representability of polynomial function, specular highlight and shadow may not be effectively represented. Extra procedure, such as estimation of surface normal, is needed to compensate the missing highlight. Moreover, its inability to handle discontinuous shadows may cause problems, as shadowing is unavoidable in most scenes. In this paper, we try to compress the apparent pixel BRDF while preserving its representability of specular highlights and shadows. No assumption on the apparent pixel BRDF is made. Our proposed method is *scalable*. It achieves higher image quality when more storage is allowed. The trade-off is that our method may not achieve the compression ratio as high as the polynomial function. Nevertheless, the final data size we achieved is only a few megabytes. Note that, in our case, specular highlight and shadow are preserved without extra procedure.

III. AN ILLUMINATION-ADJUSTABLE IMAGE REPRESENTATION

For the completeness of this paper, we first give a brief description of the illumination-adjustable image representation previously proposed. The details can be found in [3].

A. Plenoptic Illumination Function

Adelson and Bergen [9] proposed a seven-dimensional *plenoptic function* for evaluating the low-level human vision models. Although its original formulation is very general, the illumination and other scene changing factors are embedded inside a single time parameter. Techniques [8] [1] [11] [10] based on this model also inherit this rigidity. Most of them assumed the

illumination is unchanged and the scene is static. We extended the plenoptic function to support relighting by extracting the illumination component (\vec{L}) from the aggregate time parameter and explicitly specifying it in the following new formulation,

$$I = P_I(\vec{L}, \vec{V}, \dot{E}, t', \lambda), \quad (1)$$

where I is the radiance; \vec{L} specifies the direction of a directional light source illuminating the scene; \vec{V} specifies the viewing direction; \dot{E} is the position of viewpoint; t' is the time parameter after extracting the illumination component; and λ is the wavelength.

We call this new formulation the *plenoptic illumination function*. Intuitively speaking, the function tells us how the environment looks like when illuminated by a directional light source. The difference between our formulation and the reflectance field [29] is mainly on the parameterization. While our parameterization is viewpoint-centric, their definition is object-centric. Hence our definition supports image-based scenes with multiple objects.

B. Sampling and Relighting

Sampling the plenoptic illumination function is actually a process of taking pictures. The question is how to take those pictures. The time parameter t' (the scene) is usually assumed fixed and the wavelength parameter λ is conventionally sampled and reconstructed at three positions (red, blue and green). As we are interested in constant-viewpoint images, parameter \dot{E} is also fixed. The sampling of \vec{V} depends on the projection manifold we used. The manifold can be planar, cylindrical or spherical. If we consider the radiance values received through a single pixel window (\vec{V} is unchanged), the plenoptic illumination function reduces to a spherical function of \vec{L} .

Since the newly introduced dimension \vec{L} is a direction, we sample it on a spherical grid (Figure 1(a)). That is, for each grid point, we take a picture with the light source specified in the corresponding direction. In other words, for each pixel window, we obtain multiple radiance values, each value corresponds to a distinct lighting direction. Figure 1(b) shows a 3D plot of radiance values (R channel only) associated with a single pixel window collected from the set of captured images.

Given a desired light vector which is not one of the samples, the desired image can be estimated by interpolating samples. Instead of interpolating in the spatial domain, we interpolate

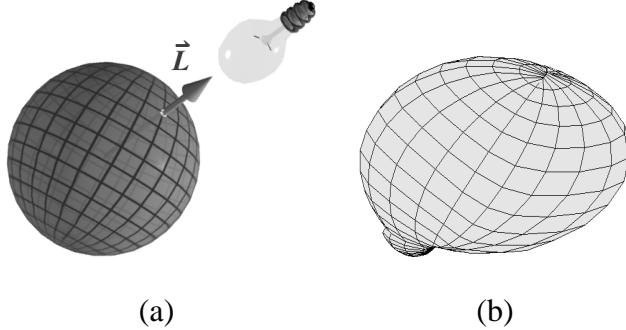


Fig. 1. (a) Sampling the lighting direction on the spherical grid. (b) A 3D plot of radiance values associated with a pixel.

the coefficients in the spherical harmonic domain (described in Section IV).

Even though the sampled plenoptic illumination function only tells us how the environment looks like when it is illuminated by a single white directional source, other illumination configurations can be simulated by making use of the linearity property of light. The relit image can be synthesized by computing the following formula for each pixel and for each red, green and blue channel,

$$\sum_i^n P_I^*(\theta_l^i, \phi_l^i) L_r(\theta_l^i, \phi_l^i), \quad (2)$$

where n is the total number of light sources; (θ_l^i, ϕ_l^i) specifies the desired lighting direction, \vec{L}_i , of the i -th light source; $P_I^*(\theta_l^i, \phi_l^i)$ is the result of interpolating the samples given the desired light vector (θ_l^i, ϕ_l^i) . The parameters \vec{V} , \vec{E} , t' and λ are dropped for simplicity; L_r is the radiance along (θ_l^i, ϕ_l^i) due to the i -th light source.

This formula tells us the radiance value coming along certain viewing direction \vec{V} under a desired illumination configuration which may consist of multiple light sources. P_I^* is the estimated radiance by interpolating the samples. This formula allows us to manipulate three parameters, namely the direction, the color, and the number of light sources. Figure 2 shows some results of applying Equation (2) with different types of light sources. Relighting with non-directional light sources, such as in Figures 2(a) & (c)-(e), requires the knowledge of depth value in order to correctly compute the light vector [2]. Figure 2 shows that relighting can be performed on panoramic images as well as perspective images.

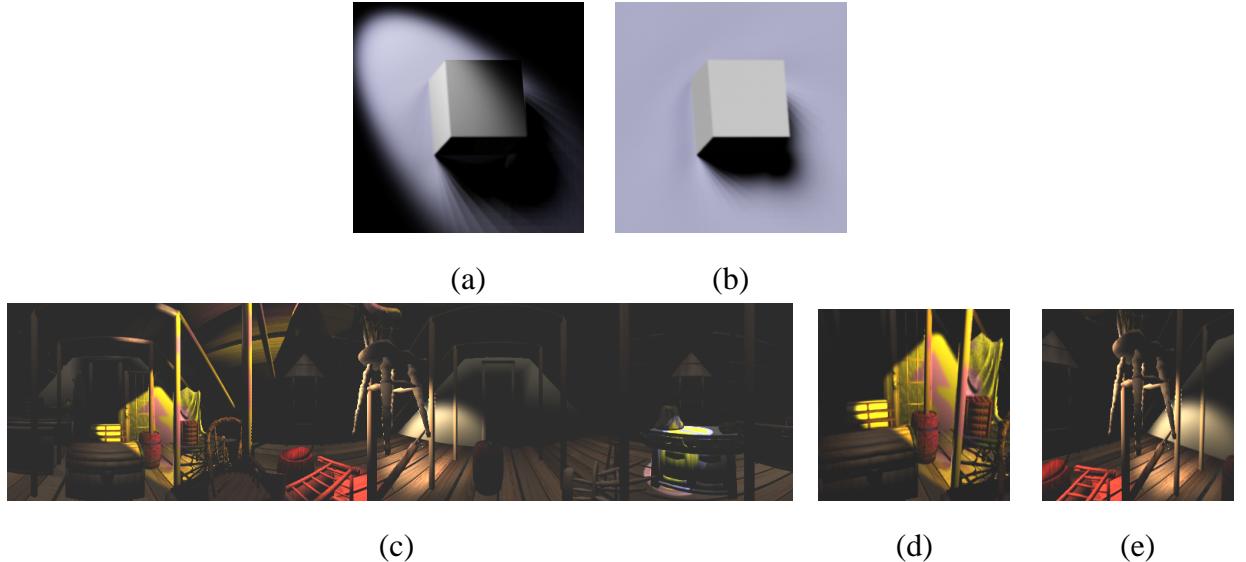


Fig. 2. Relighting by (a) a spotlight and (b) a directional light source. (c) Relighting can be performed on panoramic images as well as perspective images. (d) & (e) show the perspective snapshots.

IV. INTRA-PIXEL COMPRESSION

Consider a pixel in an illumination-adjustable image, there must be an associated viewing ray which connects the viewpoint to the pixel window. If the viewpoint, the viewing direction and the scene are all static during the data capture and only the illumination is allowed to change, it is very likely that radiance values captured under various lighting conditions are strongly correlated because most geometric factors are frozen. Geometry is usually the major source of discontinuity of radiance values [30]. Moreover the received radiance values are strongly related to the surface reflectance of the visible surface element. However, they are not the same because there exists discontinuity in the received radiances due to the shadow cast by the nearby geometry.

We first group together those radiance values related to the same pixel window (or the same viewing ray) and try to make use of this *intra-pixel data correlation*. The terminology “*intra-pixel*” refers to the data correlation associated with the abstract *pixel* window. This correlation is regarded as the inter-image correlation in the traditional video coding approaches as we group values from adjacent images.

The grouped radiance values are indexed only by the light vector \vec{L} or (θ_l, ϕ_l) . In other words, it is a spherical function. Storing this spherical function without any compression requires huge

storage. The actual storage requirement depends on the sampling rate of the spherical grid. If the sampling rate is 30×40 along dimensions θ and ϕ respectively. This implies 1,200 color values (3 bytes each) or 3.5KB are needed for each pixel. A 256×256 image hence requires 225MB of storage which is obviously not practical.

A. Spherical Harmonics

To compress them, we transform this spherical function into the spherical harmonic domain [31], and the resultant spherical harmonic coefficients are zonally sampled and quantized (Figure 3). The spherical harmonic transform has been used for compressing the BRDF [32] in various previous works [33], [34]. The spherical harmonic transform are summarized as follows.

$$C_{l,m} = \int_0^{2\pi} \int_0^\pi P_I(\theta, \phi) B_{l,m}(\theta, \phi) \sin \theta d\theta d\phi, \quad (3)$$

where $P_I(\theta, \phi)$'s are the sampled radiance values,

$$B_{l,m}(\theta, \phi) = \begin{cases} N_{l,m} Q_{l,m}(\cos \theta) \cos(m\phi) & \text{if } m > 0 \\ N_{l,0} Q_{l,0}(\cos \theta) / \sqrt{2} & \text{if } m = 0 \\ N_{l,m} Q_{l,m}(\cos \theta) \sin(|m|\phi) & \text{if } m < 0, \end{cases}$$

$$N_{l,m} = \sqrt{\frac{2l+1}{2\pi} \frac{(l-|m|)!}{(l+|m|)!}},$$

and

$$Q_{l,m}(x) = \begin{cases} (1-2m)\sqrt{1-x^2} Q_{m-1,m-1}(x) & \text{if } l = m \\ (2m+1)x Q_{m,m}(x) & \text{if } l = m+1 \\ \frac{2l-1}{l-m} x Q_{l-1,m}(x) - \frac{l+m-1}{l-m} Q_{l-2,m}(x) & \text{otherwise.} \end{cases}$$

and $Q_{0,0}(x) = 1$.

Coefficients $C_{l,m}$ are the spherical harmonic coefficients to be zonally sampled, quantized and stored. Functions $Q_{l,m}(x)$ are the Legendre polynomials. From Equation (3), the integration can be computed easily if samples are taken on the spherical grid points. This is also why we choose to sample on the spherical grid.

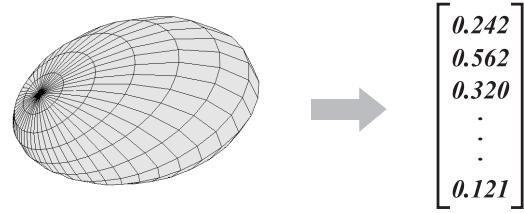


Fig. 3. Spherical harmonic transform.

Intuitively speaking, the spherical harmonic transform can be regarded as a Fourier transform in the spherical domain. Just like the Fourier transform, the more coefficients are used for representation, the more accurate the reconstructed value is. Figure 4 shows the first few harmonics (basis functions $B_{l,m}(\theta, \phi)$). Besides the first basis function (which is a sphere), all other basis functions exhibit directional preferences. Different input functions should exhibit differences in the size of harmonics after transform. Note that basis functions may return negative values. Negative lobes are visualized through “flipping” about the origin in Figure 4. In some cases, these negative lobes may coincide with positive lobes. The two basis functions on the second row of Figure 4 actually contain two lobes, one negative and one positive. However, the negative lobe coincides with the positive lobe.

To reconstruct an interpolated radiance value, the following summation of multiplications is calculated,

$$P_I^*(\theta, \phi) = \sum_{l=0}^{l_{\max}} \sum_{m=-l}^l C_{l,m} B_{l,m}(\theta, \phi). \quad (4)$$

where $(l_{\max} + 1)^2$ is the number of spherical harmonic coefficients stored.

B. Visual Evaluation

In general, the more spherical harmonic coefficients are used for representation, the more accurate the approximation is. Figure 5 shows the visual effects when different number of coefficients are kept for encoding. Comparing to the original reference image in Figure 5(d), the glossy teapot becomes diffuse when fewer coefficients are used (Figures 5(a)-(c)). This can be explained by the shape of harmonics in Figure 4. The low-order spherical harmonics are responsible for the diffuse component as the diffuse component is a smooth function. On the other hand, the higher-order harmonics are responsible for directional, specular highlight and shadow.

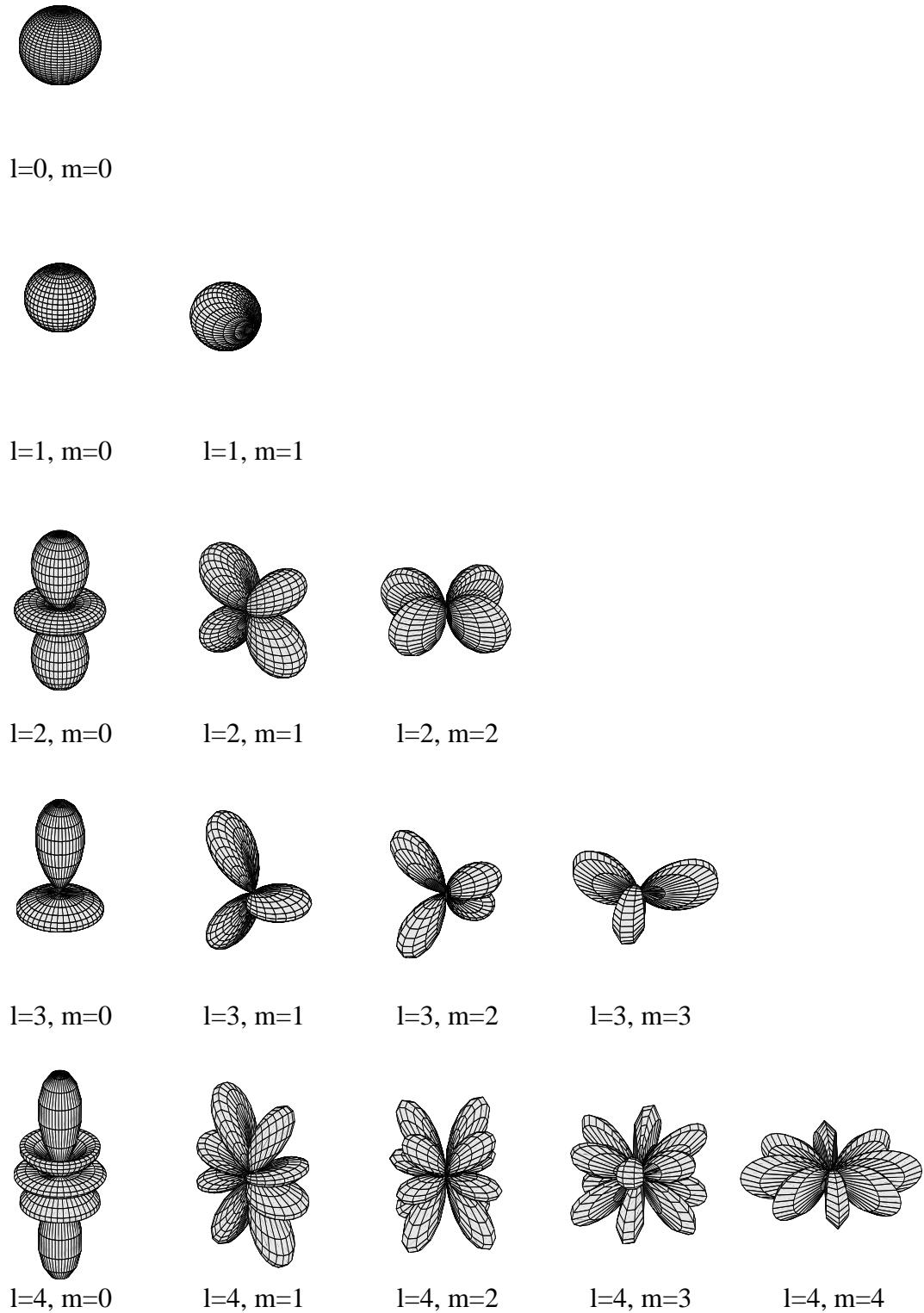


Fig. 4. Spherical harmonics. Negative lobes may coincide with positive lobes such as $B_{1,0}$ and $B_{1,1}$.

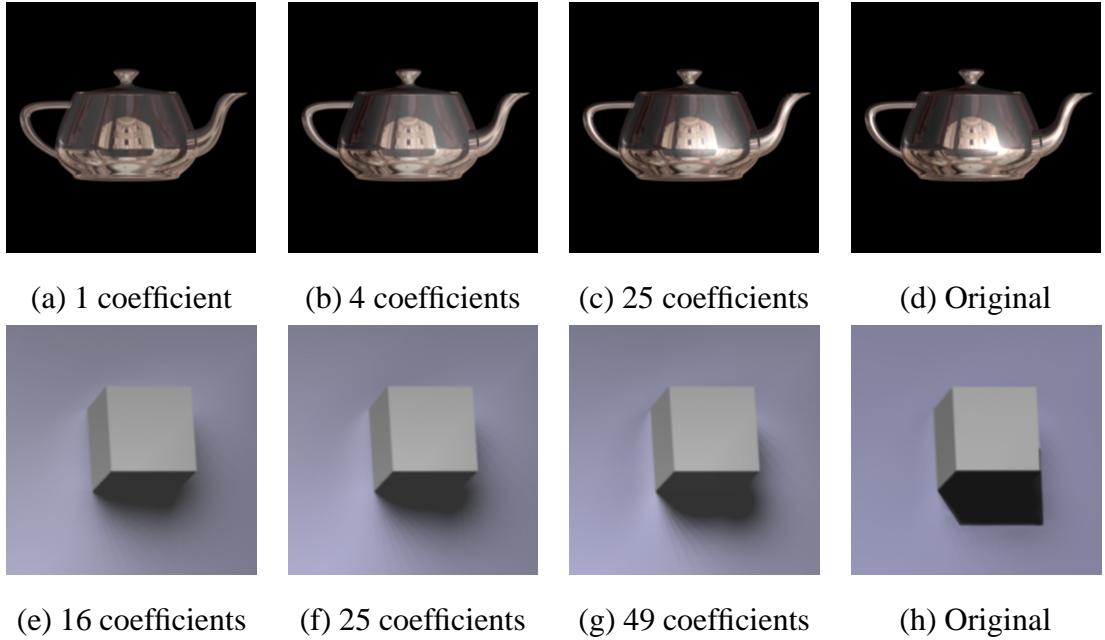


Fig. 5. The visual effect of using different numbers of spherical harmonic coefficients for encoding. (a)-(c): Specular highlight and (e)-(g): shadow accuracy. (d) & (h): Original reference images.

Therefore, truncating higher-order coefficients should result in images with more diffuse appearance. When there is shadow, dropping higher-order coefficients also reduces the accuracy of the reconstructed images (Figure 5(e)-(g)). Since the shadow introduces discontinuity to the plenoptic illumination function, it requires infinite number of coefficients to represent.

The optimal (in term of image quality) number of spherical harmonic coefficients used for compression depends on the image content. Images containing specular object require more coefficients than images with only diffuse object. Images containing shadow also require more coefficients to represent than ones without shadow. In most of our tested scenes, 25 coefficients provide a good balance between storage and visual quality.

C. Statistical Evaluation

To objectively evaluate the quality of the spherical harmonic compression, we measure the peak signal-to-noise ratio (PSNR) against the number of coefficients used for encoding. The experiment is carried out in the following manner. The illumination-adjustable panorama of attic (Figure 2(c)), which contains three hundred $1,024 \times 256$ reference images, is taken as the test data. Five hundred desired images are generated from spherical harmonic coefficients by positioning a directional light source at five hundred distinct directions. Each sampled direction

is equivalent to a sample point on the spherical surface. To sample without bias, the sample points are evenly distributed on the spherical surface using Hammersley point set [35], which has been proved [36] to generate uniform and stratified sample points on a sphere.

The image-based scene can be illuminated with arbitrary type and arbitrary number of light sources. However, to investigate the noise introduced solely by the spherical harmonic compression, we only evaluate the desired images generated with a single directional light source as all reference images are captured under the illumination of a single directional light source. The control images are the rendered images generated by traditional geometry-based renderer with the geometrically represented scene as input. The same five hundred illumination conditions are applied for generating the control images.

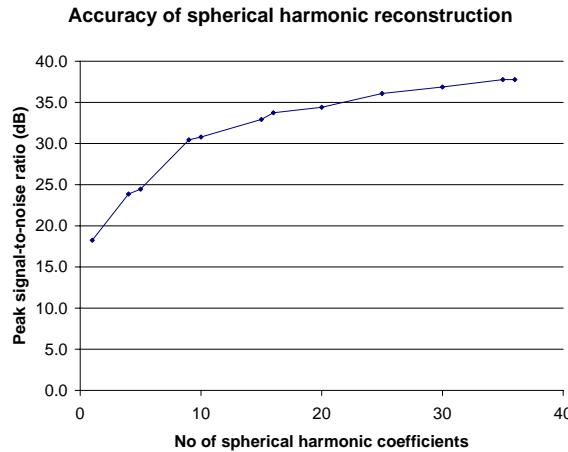


Fig. 6. PSNR vs. number of spherical harmonic coefficients used.

We measure the PSNR by taking the mean-squared-error (where error is the difference between the corresponding pixel values in the control images and the desired images) as the power of noise and the square of maximum pixel value as the power of signal. Figure 6 shows the average PSNR (in decibel) against the number of spherical harmonic coefficients used for encoding.

From the figure, the PSNR consistently improves with the increase in the number of spherical harmonic coefficients. However, the improvement of PSNR slows down after 25 coefficients. Hence the cost-effective number of coefficients used for the attic scene is around 25. Similar statistics are found for other scenes we have tested. This also justifies our claim that 25 coefficients are usually sufficient for most scenes.

Since the number of spherical harmonic coefficients kept is almost linear to the total storage requirement (explained in Section V), the curve in Figure 6 also reveals the relationship between the PSNR and the total storage.

V. INTER-PIXEL COMPRESSION

Up to now, we only utilize the correlation among radiance values received through the same pixel window. The data correlation between adjacent pixels has not yet been exploited. We called it the *inter-pixel* correlation as it refers to the correlation between adjacent abstract pixels. Note that this kind of correlation is instead regarded as the intra-image correlation in standard image coding.

Since radiance values associating with a pixel are now transformed to a coefficient vector, we have to exploit the correlation between these adjacent coefficient vectors. One may suggest to compress the data using vector quantization (VQ) techniques with the coefficient vectors as input. However, a well-known visual artifact of VQ techniques is *contouring* at the region with smooth changes. Although dithering can partially reduce the objectionable visual artifact, there is an even more difficult problem, large computational expense. It is well known that the computational time of VQ techniques is substantially high when the input data size is large, which is in our case. Moreover, with the same tolerance of reconstruction error, the compression ratio obtained by VQ techniques is usually lower than that of other methods. All these disadvantages prohibit us from using VQ techniques.

Instead, we pick the first coefficients (Figure 7) from all coefficient vectors and form a coefficient map. To avoid terminology ambiguity, this kind of coefficient map is called *SH map* (*SH* stands for Spherical Harmonics) from now on. The same grouping process is applied to the second coefficients and all other coefficients. The result is a set of k SH maps if the coefficient vectors are k -dimensional. Interestingly, each SH map is in fact an image (right hand side of Figure 7). These SH maps are somewhat analogous to the basis images found by the principal component analysis in the computer vision literatures [20] [22]. This observation suggests us a way to utilize the *inter-pixel correlation*. We can simply treat each SH map as an image of real values and apply standard image compression. To do so, we apply the discrete wavelet transform (DWT) [37].

It seems that different amount of storage (bits) should be allocated to different SH maps based

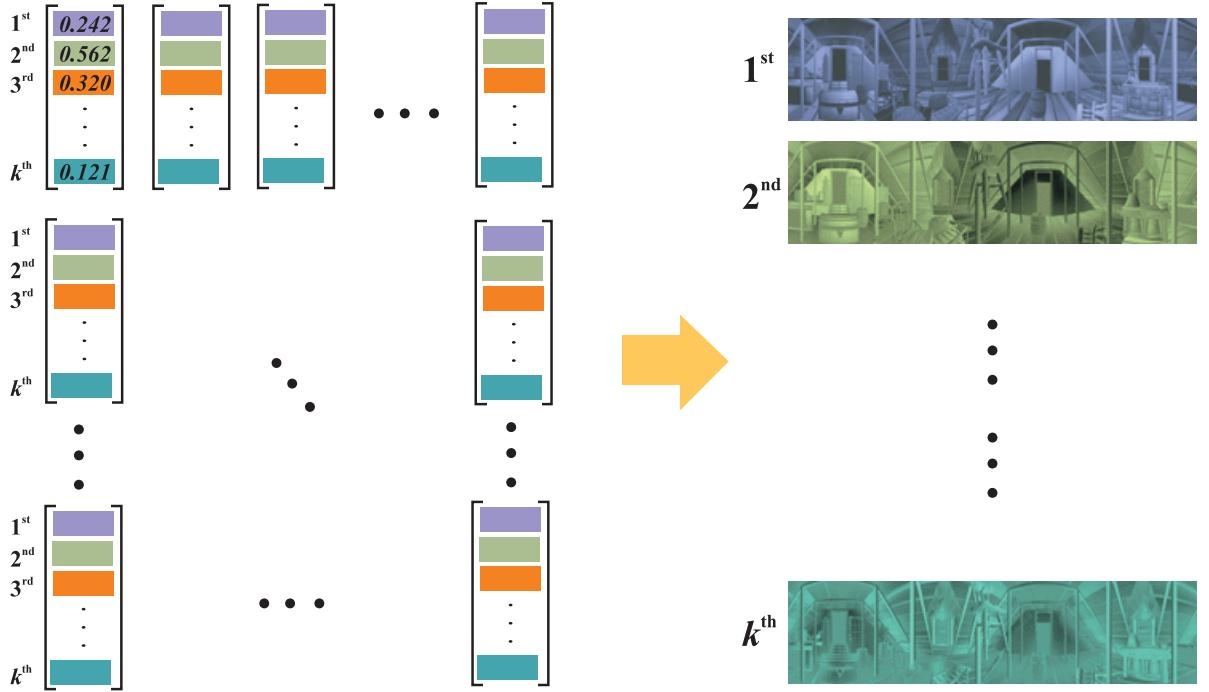


Fig. 7. Picking the coefficients from the vectors and forming the SH maps.

on the data statistics. For example, more bits should be assigned to low-order SH maps than other higher-order SH maps. However, based on standard bit allocation methods, low-order SH maps occupy too many bits while high-order SH maps starve (of bits). It is expected as low-order SH maps contribute more energy to the reconstructed images than the high-order SH maps do. As a result, specular highlight and shadow are poorly preserved or even lost since they are represented by high-order SH maps. To preserve these human-sensitive visual features, we assign the same bit rate to each SH map in the same color channel to prevent high-order SH maps from starvation. This explains why the total storage is almost linearly proportional to the total number of SH maps. Note that SH maps in different color channels may receive different bit rates (described shortly in Section VI).

A. Discrete Wavelet Transform

The wavelet technique for encoding SH maps described below is very similar to the one for encoding natural images proposed by Joshi *et al.* [37]. Each SH map is decomposed by the 9-7 tap bi-orthogonal wavelet decomposition. Instead of decomposing a SH map into several levels, only two levels of decomposition is performed.

After this decomposition, each SH map is partitioned into 16 uniform bands. One of these band is the lowest frequency subband (LFS) while the other 15 bands are high frequency subbands (HFS). Figure 8 shows the 16 resultant subbands. The dark gray region is the LFS while the light gray regions are HFS's. If the size of an input SH map is $x \times y$, the size of each band will be $\frac{x}{4} \times \frac{y}{4}$.

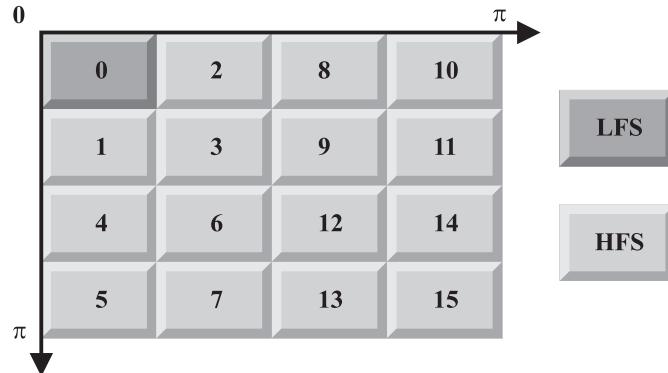


Fig. 8. After the two-level wavelet decomposition, the coefficients in the frequency domain can be partitioned into 16 subbands.

From our observation, the LFS is similar to a downsampled version of the input SH map. In other words, it is also an image. Hence we can apply the common approach for compressing the data in this subband. The LFS is first subdivided into 4×4 non-overlapping blocks. The discrete cosine transform (DCT) [38] is then applied to each of these blocks. The DCT itself does not reduce the data volume, the later quantization does. To facilitate the quantization, 16 data subsources are formed (one for each DCT coefficient in the transformed 4×4 block).

On the other hand, a different approach is applied to the 15 HFS's left. It is well known that after the wavelet decomposition, there is only small intra-band correlation among the data within each HFS. Thus we assume that each HFS is a memoryless data source.

Consider a SH map of the size $x \times y$. There will be 16 data subsources from the LFS (after DCT), each subsource contains $xy/256$ samples. For the HFS's, there will be 15 data subsources (one for each HFS), each subsource contains $xy/16$ samples.

B. Quantization and Bit Rate Allocation

Each source is quantized by the generalized Lloyd algorithm [39]. The bit rate of each source is given by [37],

$$R_i = R + 0.5 \log_2(\alpha_i \sigma_i^2) - \left\{ \sum_{j=1}^M 0.5 p_j \log_2(\alpha_j \sigma_j^2) \right\}.$$

where R is the target bit rate; R_i is the encoding rate of the i -th source in bits/sample; p_i is the normalized weight for the source; σ_i^2 is the variance of the source; α_i is a user-defined constant; and M is the number of sources.

Intuitively speaking, more bits should be allocated to a source with a larger variance. Constant α_i depends on the density of source as well as the type of encoding method used. In this paper, we assume that it is independent of the encoding rate. The DC component of the DCT in the LFS is assumed to have a Gaussian distribution as described in [40]. All other sources are modeled as Laplacian sources. Constant α_i is chosen to be 2.7 and 4.5 for Gaussian and Laplacian sources respectively [41].

C. Statistical Evaluation

To evaluate the error introduced by the described discrete wavelet transform, we study the PSNR when various bit rates are considered. The experiment is carried out in the following manner. The attic panorama in Figure 2(c) is chosen as the test data. Five hundred desired images of different lighting conditions are generated just like the experiment in Section IV-C. The only difference is that the control images are not the rendered images, but the desired images generated from a data set represented by 25 spherical harmonic coefficients *without* wavelet compression. This can avoid mixing up the distortion introduced by the spherical harmonic transform. With this set of control images, we can investigate the noise introduced solely by the wavelet compression.

We then compare the difference between the control images and the images generated from wavelet-compressed data sets with different bit rates. The smaller the bit rate is, the higher the compression ratio is and in general the poorer the image quality is. Figure 9 shows the plot of PSNR against the bit rate (bits / SH coefficient). The PSNR improves as the bit rate increases. The curve demonstrates that the wavelet compression performs quite well. In fact, bit rates of 4.0 or above are seldom necessary.

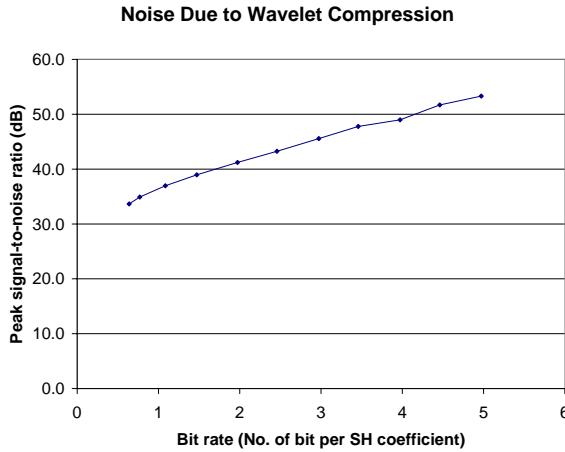


Fig. 9. PSNR vs. bit rate for wavelet compression.

D. Visual Evaluation

Figure 10 visually compares the images generated from the compressed data with various bit rates. As shown in Figure 10, the quality of the reconstructed images improves as the bit rate increases. Noise is observable at the high contrast region in the 0.5-bit image (Figure 10(b)). Unlike DCT encoded image (such as JPEG), there is no blocky artifact in all images. The artifact reduces significantly in the 1.0-bit image. There is almost no significant visual difference between the 2.0-bit and the original image.

VI. INTER-CHANNEL COMPRESSION

The final compression is done on the wavelength dimension of the plenoptic illumination function. Representing a color in the RGB space is usually wasteful in terms of storage, because the RGB model does not pack important visual signal efficiently. A common color model used in PAL video coding and JPEG standard is YUV. The major advantage is that it packs important visual energy into a single luminous channel Y. The other two channels, U and V, are chrominous channels. Experiment [42] shows that human perception are less sensitive to signal in chrominous channels than that in the luminous channel. Therefore, it is desirable to reduce the bits allocated for U and V in order to trade for an increase in bits allocated for Y.

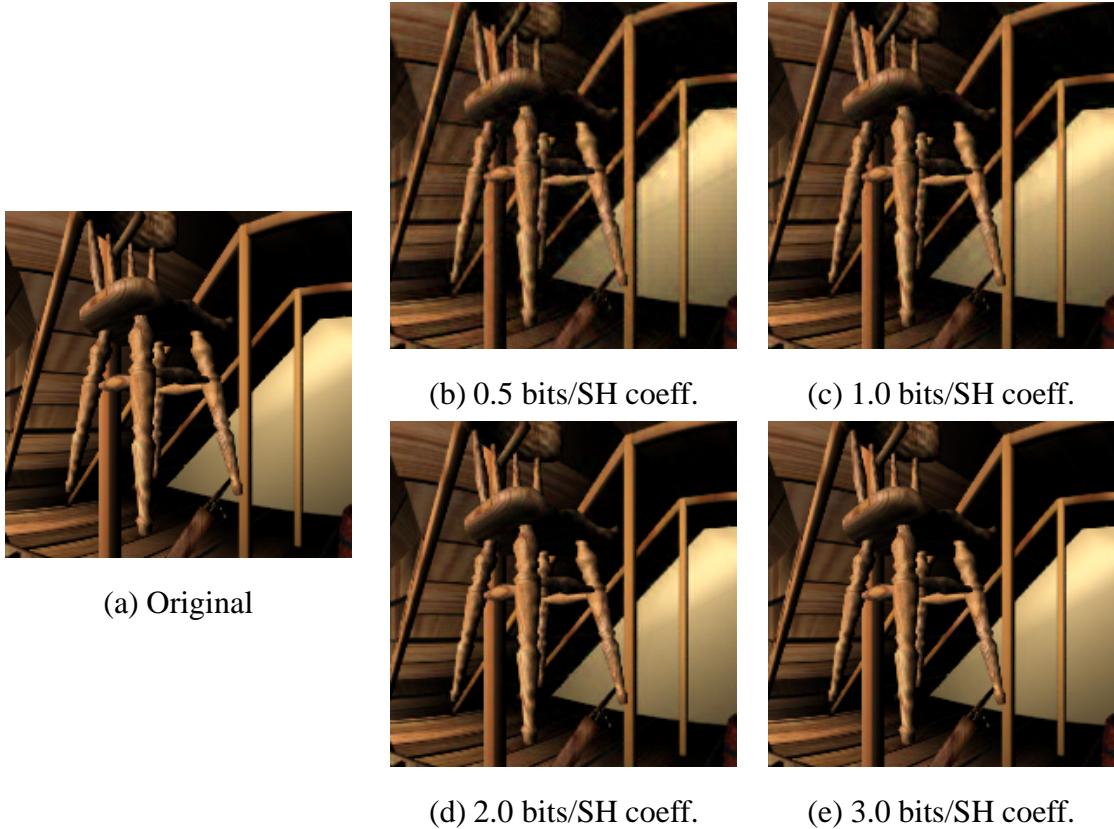


Fig. 10. Visual comparison of images encoded with various bit rates.

A. Color Conversion in Frequency Domain

A naive approach to adopt the YUV color model is to convert the RGB values of each pixel in the input reference images to YUV before passing the values to the spherical harmonic transform. However, this approach is very inefficient as a matrix multiplication is needed for each radiance value and there may be thousands of reference images.

In fact, the color conversion can be done in the spherical harmonic domain. That is, we can perform color conversion after the spherical harmonic transform. Since there are only a few spherical harmonic coefficients (say 16 to 25) for each pixel, the color conversion can be done efficiently.

The question is whether the color conversion in the spherical harmonic domain is correct. Since the color transform is linear, the correctness of this approach can be proved by the following derivation. Recall spherical harmonic transform in Equation (3).

$$C_{l,m}^Y = \int_0^{2\pi} \int_0^\pi P_I^Y B_{l,m} \sin \theta d\theta d\phi \quad (5)$$

where $C_{l,m}^Y$ is the spherical harmonic coefficient for the Y channel; P_I^Y is the radiance value in the Y channel.

The parameter (θ, ϕ) of P_I and $B_{l,m}$ is dropped for simplicity and clarity. From the PAL video standard, we have,

$$Y = \beta_1 R + \beta_2 G + \beta_3 B \quad (6)$$

where Y is the value in the Y channel; R, G, B are pixel values in R, G & B channels respectively; $\beta_1 = 0.299$, $\beta_2 = 0.587$ and $\beta_3 = 0.114$.

Substituting Equation (6) into Equation (5),

$$\begin{aligned} C_{l,m}^Y &= \int_0^{2\pi} \int_0^\pi (\beta_1 P_I^R + \beta_2 P_I^G + \beta_3 P_I^B) B_{l,m} \sin \theta d\theta d\phi \\ &= \beta_1 \int_0^{2\pi} \int_0^\pi P_I^R B_{l,m} \sin \theta d\theta d\phi + \beta_2 \int_0^{2\pi} \int_0^\pi P_I^G B_{l,m} \sin \theta d\theta d\phi \\ &\quad + \beta_3 \int_0^{2\pi} \int_0^\pi P_I^B B_{l,m} \sin \theta d\theta d\phi \\ &= \beta_1 C_{l,m}^R + \beta_2 C_{l,m}^G + \beta_3 C_{l,m}^B \end{aligned}$$

where P_I^R, P_I^G, P_I^B are pixel values in R, G & B channels; $C_{l,m}^R, C_{l,m}^G, C_{l,m}^B$ are spherical harmonic coefficients in R, G & B channels.

Similar derivation can be done for channels U and V. Hence, the color conversion can be applied to spherical harmonic coefficients in the frequency domain instead of the original spatial domain.

B. Statistical Evaluation

To evaluate the effectiveness of the color transform, we compare the PSNR of data encoded in the YUV space with that in the RGB space. Each time the same *average* bit rate is assigned to both data sets. For the RGB data set, the ratio of bits allocated to R, G and B SH maps is 1:1:1. For the YUV data set, the ratio of bits allocated to Y, U and V SH maps is 2:1:1. For instance, if an average bit rate of 2 is wanted, each coefficient in Y, U and V channels should receive 3, 1.5

and 1.5 bits respectively. We decide to allocate more bits to U and V channels than that in PAL standard. The basic idea is not changed, *i.e.* more bits for the luminous Y channel.

From each data set, 500 desired images are generated just like before. The control images are same as those in the previous experiment in Section V-C. The average PSNR is then plotted against the bit rate. Figure 11 shows the curves of YUV and RGB data sets. The YUV data set in general preserves better image quality than the RGB data set, given the same compression ratio. The difference in PSNR becomes apparent as the bit rate increases.

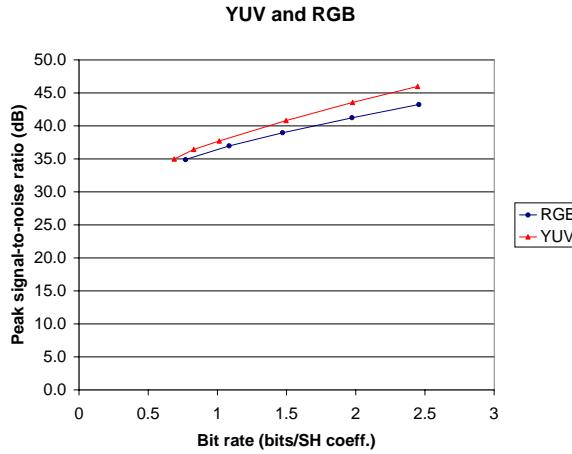


Fig. 11. PSNR vs. bit rate using YUV and RGB color models.

C. Visual Evaluation

Although the statistical evaluation shows that PSNRs of YUV and RGB data sets are very similar at low bit rates such as 0.5 and 1.0, some visual artifacts are not revealed in the statistics. Figure 12 gives the visual comparison of the results from RGB and YUV data sets. The two images from RGB data set on the left column exhibit some observable red and green “clouds” in the smooth region. This kind of artifact is more apparent when the bit rate is low (Figure 12(a)). No similar artifact is observable at the same place in the YUV results.

VII. OVERALL EVALUATION

To evaluate the overall performance of the proposed method, we compare it to standard image compression methods, JPEG, JPEG2000, and MPEG. JPEG is a standard compression method for still images. The wavelet-based JPEG2000 [43] is a new image coding standard which

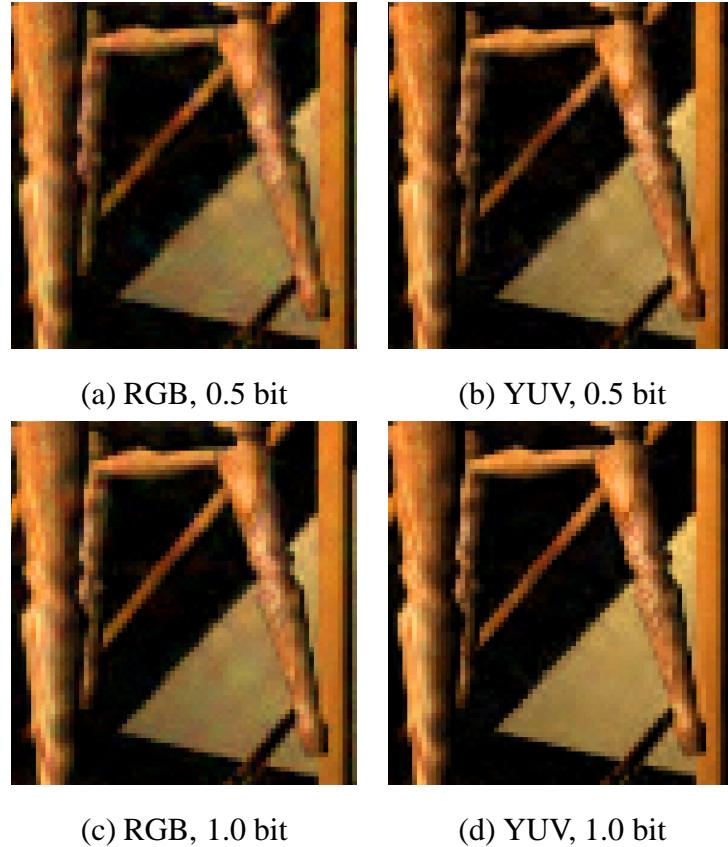


Fig. 12. Visual comparison of RGB and YUV results.

Data set	Resolution	Sampling rate($\theta \times \phi$)	Real/Synthetic	With Shadow
attic	1024×256	15×20	Synthetic	No
forbid	1024×256	30×40	Synthetic	Yes

TABLE I

CHARACTERISTICS OF TESTED DATA SETS.

performs better than JPEG at low bit rates. However, both JPEG and JPEG2000 exploits no inter-image correlation. Hence we also compare our method to MPEG which is designed for video coding. Two synthetic data sets, ‘attic’ and ‘forbid’, are tested (Figure 13). The ‘attic’ data set contains three hundred 1024×256 reference images while the ‘forbid’ data set contains one thousand and two hundred 1024×256 reference images. The ‘attic’ data set contains specular objects while the data set ‘forbid’ contains shadow. Table I shows the data property of each data set.

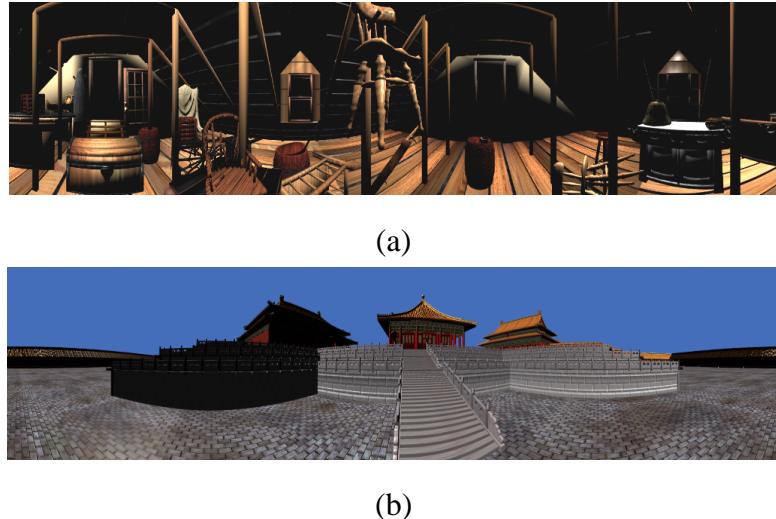


Fig. 13. Two data sets tested throughout the experiment: (a) ‘attic’, and (b) ‘forbid’. Both are synthetic.

When compressing the data sets with our method, we keep the number of spherical harmonic coefficients to be 25 and use the YUV color model. The bit rate allocation for Y:U:V channels is 2:1:1 in ratio. The only parameter to vary is the target bit rate for wavelet compression. Then we reconstruct images from the compressed data at the sampling positions. The control images are the original reference images. The same input image set is passed to JPEG, JPEG2000 and MPEG for compression. Two graphs of PSNR versus bits per pixel are plotted, one for ‘attic’ (Figure 14(a)) and the other for ‘forbid’ (Figure 14(b)). The “pixel” in the “bits per pixel” refers to the original image pixel (each pixel stores RGB values, 3 bytes in size), not the abstract pixel we referred in the previous sections. The performance curves of our method, JPEG, JPEG2000 and MPEG are plotted in these two graphs. Besides, we also plot the graphs of PSNR versus compression ratio (Figure 14(c) & (d)) for reference.

From the graphs, the proposed method out-performs JPEG, JPEG2000 and MPEG. At the same PSNR level, we achieve a much lower bit rate (or higher compression ratio) than other three coding standards, especially at low bit rates. Note that, since the size of our data size is *always enormous*, it is meaningless to perform well at high bit rates. Consider the case of ‘forbid’, the total data size is 900MB, compressing with 0.3 bit per pixel implies a storage of 90 MB is required. This is still too much for Internet-based applications. Hence, one major criterion for our codec is that it must perform extremely well at low bit rates. From the statistics, our method satisfies such requirement.

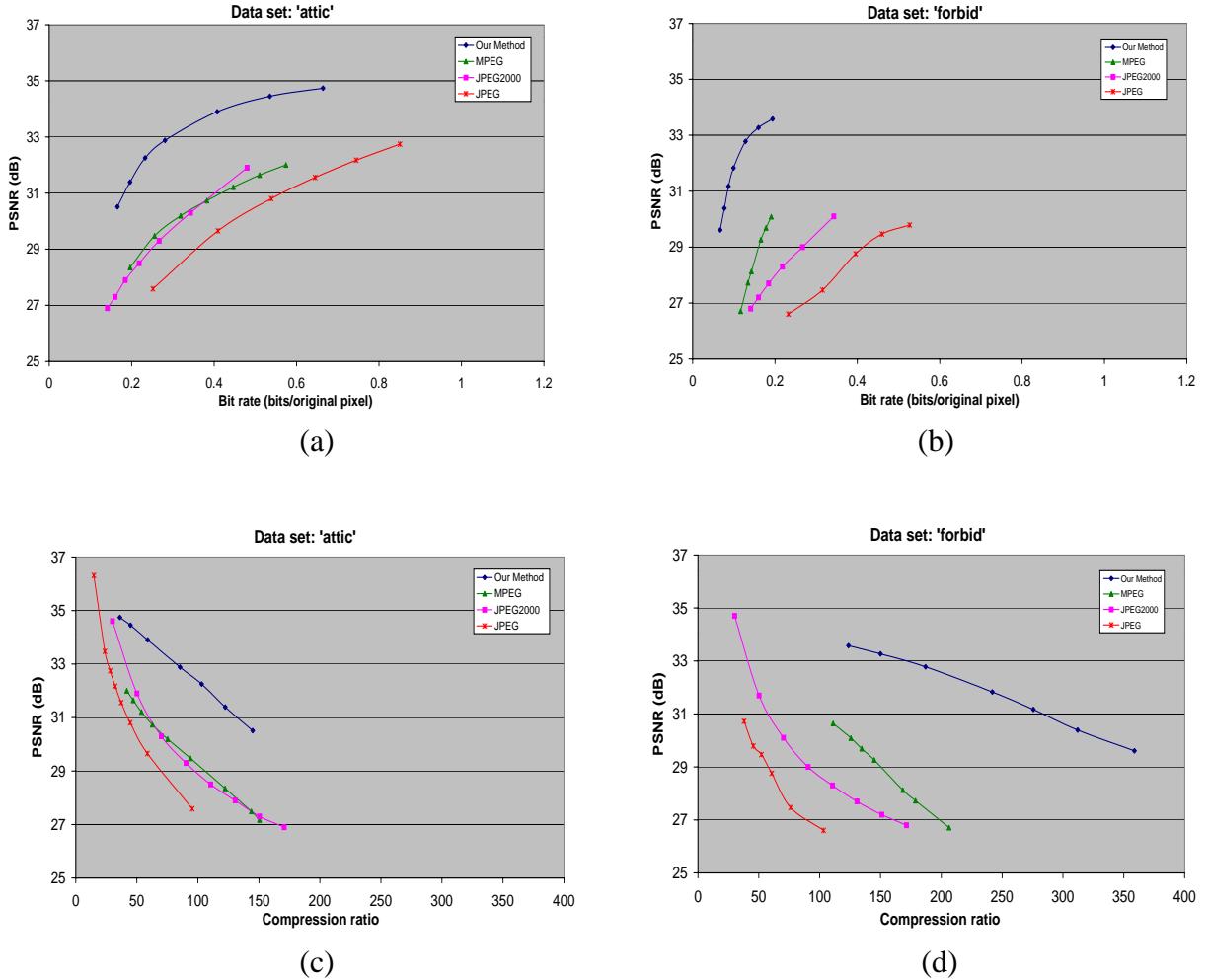


Fig. 14. Performance comparison with JPEG, JPEG2000 and MPEG. (a) & (b): PSNR vs. Bit per original pixel. (c) & (d): PSNR vs. Compression ratio.

Among the compared methods, JPEG is the worst. JPEG2000 performs better than JPEG, especially at low bit rates. In the case of ‘attic’, JPEG2000 is comparable to MPEG. However, both JPEG and JPEG2000 are inferior to ours because they do not exploit the correlation among images. As MPEG only exploits the correlation in 1D (time domain), it may not fully utilize the correlation in our data set which is a set of samples on the 2D spherical surface. It might also be because MPEG is tailored for object motion in video, but not for change of lighting in our case.

Figures 14(a) and (b) show that our method saturates after certain points. The reconstruction error are strongly related to the number of spherical harmonic coefficients kept. From Figure 6, the PSNR of ‘attic’ data is around 35 dB when 25 coefficients are kept (without further compres-

sion). As we use 25 coefficients in the overall performance evaluation, the PSNR we obtained should not be higher than this value. Using our method, the sizes of data with PSNR of around 30 dB are roughly 1.5 MB and 6 MB for ‘attic’ and ‘forbid’ respectively. Obviously, these compressed files can be rapidly transferred through Internet nowadays.

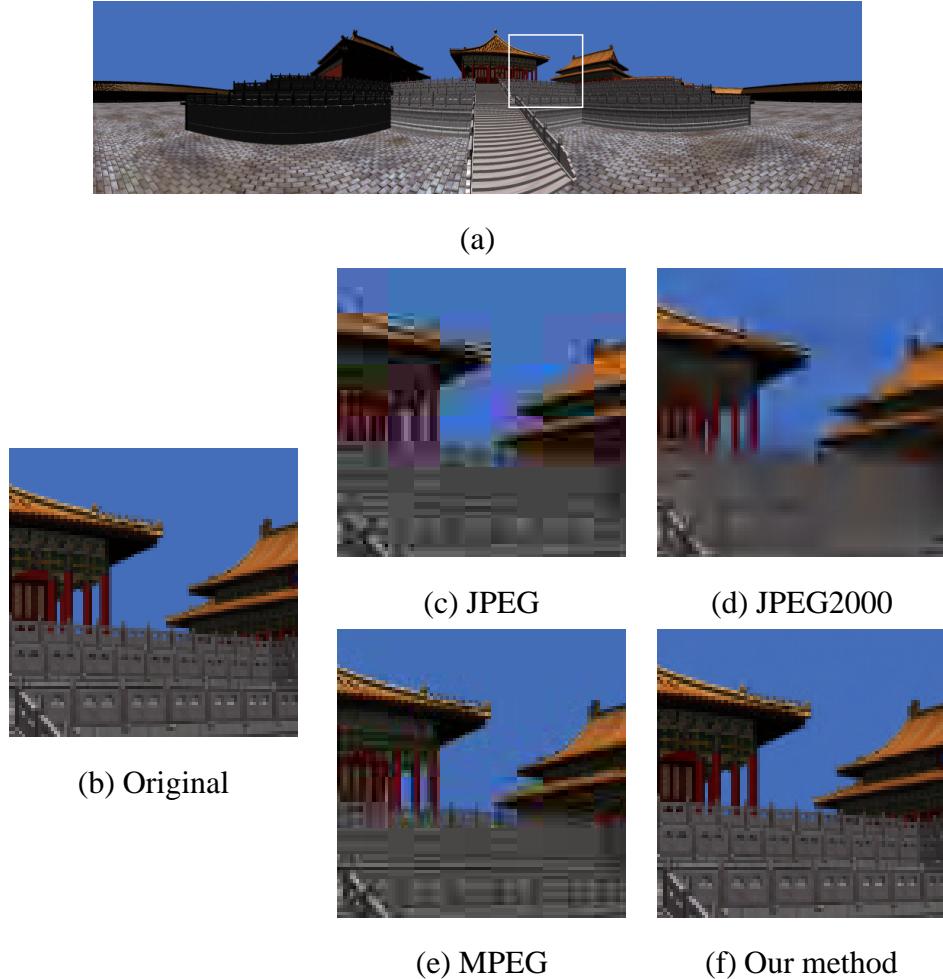


Fig. 15. Visual comparison of the proposed method, JPEG, JPEG2000 and MPEG.

Besides, we also visually compare the images compressed by our method to that of JPEG, JPEG2000 and MPEG. We fix the compression ratio at around 120 and recover an image from the data compressed by four methods. Figure 15 shows the comparison. A region in the reconstructed image (Figure 15(a)) is blowed up for comparison. The JPEG result (Figure 15(c)) contains serious blocky and blurry artifacts. The color tones of some blocks are even distorted. Although the JPEG2000 result (Figure 15(d)) is visually better than that of JPEG, it is too blurry that details (e.g. fence) are completely lost. The MPEG one (Figure 15(e)) is better but is still

contaminated with blocky artifact. Note that the details of fence cannot be clearly observed. The image coded by our method (Figure 15(f)) exhibits the minimal visual artifact. Comparing to the original image in Figure 15(b), there are some differences near the high contrast region. However the artifact is not apparent.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we propose a 3-step compression scheme for compressing illumination-adjustable images. A series of compression techniques is applied to exploit the intra-pixel, inter-pixel and inter-channel data correlations.

From our experiments, we show that 25 spherical harmonic coefficients are in general cost effective for intra-pixel compression. It can be shown that 1.0 to 2.0 bits per spherical harmonic coefficient is sufficient for inter-pixel compression. We also demonstrate that YUV color model should be used in compressing the color data. The proposed method out-performs standard image and video coding methods, JPEG, JPEG2000 and MPEG. A 1024×256 illumination-adjustable panoramic image sampled under 1,200 illumination configurations required only a few megabytes of storage. Note that arbitrary lighting configuration can be specified to relight from this compressed data and the time for relighting is independent of scene complexity.

One future direction to investigate is to further compress the data by exploiting more sophisticated approximation methods of BRDF, such as spherical wavelet [44] and non-linear approximation [45]. If the coefficients recovered by these methods also exhibit strong correlation, inter-pixel compression can also be applied to further reduce the storage. In this paper, we give a detail description on designing a coder for SH maps. We think that coding methods like SPIHT [46] and EBCOT [47] may also be applied after adapting to our data.

Currently, we only handle constant-viewpoint images. When the viewpoint is allowed to change, the data volume will certainly be increased. However, there should also be data correlation between images from adjacent viewpoints. There is a remarkable similarity between image-based data compression and video compression when the viewpoint is allowed to change. Further investigation is needed as the user behavior of navigating the image-based scenery is different from video playback which is quite sequential. The user behavior strongly affects the design of compression engines.

WEB AVAILABILITY

Demonstrative programs and useful resources are available through the associated web page:
<http://www.cse.cuhk.edu.hk/~ttwong/papers/plenwave/plenwave.html>.

ACKNOWLEDGMENT

This research was supported by the RGC Direct Grant for Research, the Chinese University of Hong Kong (Project ID: 2050262) and the Strategic Grant, City University of Hong Kong (Project No. 7001079).

REFERENCES

- [1] Shenchang Eric Chen, “QuickTime VR - an image-based approach to virtual environment navigation,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH’95*, August 1995, pp. 29–38.
- [2] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng, “Image-based rendering with controllable illumination,” in *Eighth Eurographics Workshop on Rendering*, Saint Etienne, France, June 1997, pp. 13–22.
- [3] Tien-Tsin Wong, Chi-Wing Fu, Pheng-Ann Heng, and Chi-Sing Leung, “The plenoptic illumination function,” *IEEE Transactions on Multimedia*, vol. 4, no. 3, pp. 361–371, September 2001.
- [4] Tien-Tsin Wong, Chi-Wing Fu, and Pheng-Ann Heng, “Interactive relighting of panoramas,” *IEEE Computer Graphics & Applications*, vol. 21, no. 2, pp. 32–41, March-April 2001.
- [5] Tien-Tsin Wong, Siu-Hang Or, and Chi-Wing Fu, “Real-time relighting of compressed panoramas,” in *Graphics Programming Methods*, Jeff Lander, Ed. Charles Rivers Media, 2003.
- [6] Paul S. Heckbert, “Survey of texture mapping,” *IEEE Computer Graphics and Applications*, vol. 6, no. 11, pp. 56–67, November 1986.
- [7] Oliver Faugeras and Luc Robert, “What can two images tell us about a third one?,” Tech. Rep., INRIA, July 1993.
- [8] Leonard McMillan and Gary Bishop, “Plenoptic modeling: An image-based rendering system,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH’95*, August 1995, pp. 39–46.
- [9] Edward H. Adelson and James R. Bergen, “The plenoptic function and the elements of early vision,” in *Computational Models of Visual Processing*, Michael S. Landy and J. Anthony Movshon, Eds., chapter 1, pp. 3–20. MIT Press, 1991.
- [10] Marc Levoy and Pat Hanrahan, “Light field rendering,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH’96*, August 1996, pp. 31–42.
- [11] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen, “The lumigraph,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH’96*, August 1996, pp. 43–54.
- [12] Heung-Yeung Shum and Li-Wei He, “Rendering with concentric mosaics,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH’99*, August 1999, pp. 299–306.
- [13] Jin-Xiang Chai, Xin Tong, Shing-Chow Chan, and Heung-Yeung Shum, “Plenoptic sampling,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH 2000*, July 2000, pp. 307–319.
- [14] Zhouchen Lin, Tien-Tsin Wong, and Heung-Yeung Shum, “Relighting with the reflected irradiance field: Representation sampling and reconstruction,” in *Proceedings of IEEE Computer Vision and Pattern Recognition 2001 (CVPR 2001)*, Hawaii, December 2001, vol. 1, pp. 561–567, IEEE Computer Society.

- [15] M. Magnor and B. Girod, “Adaptive block-based light field coding,” in *Proceedings of 3rd International Workshop on Synthetic and Natural Hybrid Coding and Three-Dimensional Imaging IWSNHC3DI’99*, Santorini, Greece, September 1999, pp. 140–143.
- [16] C. Zhang and J. Li, “Compression of lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering,” in *IEEE Data Compression Conference*, Snowbird, Utah, March 2000, pp. 253–262.
- [17] Gavin S. P. Miller, Steven Rubin, and Dulce Ponceleon, “Lazy decompression of surface light fields for precomputed global illumination,” in *Eurographics Rendering Workshop 1998*, Vienna, Austria, June 1998, pp. 281–292, Springer Wein / Eurographics.
- [18] C. Zhang and J. Li, “Compression and rendering of concentric mosaics with reference block codec (RBC),” in *Visual Communication and Image Processing (VCIP) 2000*, Perth, Australia, 2000.
- [19] G. Golub and C. van Loan, *Matrix Computations*, The John Hopkins University Press, 1989.
- [20] Peter N. Belhumeur and David J. Kriegman, “What is the set of images of an object under all possible lighting conditions,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.
- [21] Shree K. Nayar and Hiroshi Murase, “Dimensionality of illumination in appearance matching,” in *IEEE International Conference on Robotics and Automation*, April 1996, pp. 1326–1332.
- [22] Russell Epstein, Peter W. Hallinan, and Alan L. Yuille, “ 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models,” in *IEEE Workshop on Physics-based Modeling in Computer Vision*, June 1995, pp. 108–116.
- [23] Ko Nishino, Yoichi Sato, and Katsushi Ikeuchi, “Eigen-texture method: Appearance compression based on 3d model,” in *Proceedings of IEEE Computer Vision and Pattern Recognition Conference ’99*, June 1999, pp. 618–624.
- [24] Ravi Ramamoorthi and Pat Hanrahan, “An efficient representation for irradiance environment maps,” in *Proceedings of ACM SIGGRAPH 2001*. August 2001, Computer Graphics Proceedings, Annual Conference Series, pp. 497–500, ACM Press / ACM SIGGRAPH.
- [25] Ravi Ramamoorthi and Pat Hanrahan, “A signal-processing framework for inverse rendering,” in *Proceedings of ACM SIGGRAPH 2001*. August 2001, Computer Graphics Proceedings, Annual Conference Series, pp. 117–128, ACM Press / ACM SIGGRAPH.
- [26] Ravi Ramamoorthi and Pat Hanrahan, “Frequency space environment map rendering,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 517–526, July 2002.
- [27] Ravi Ramamoorthi, “Analytic PCA construction for theoretical analysis of lighting variability in images of a lambertian object,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 2002.
- [28] Tom Malzbender, Dan Gelb, and Hans Wolters, “Polynomial texture maps,” in *Proceedings of ACM SIGGRAPH 2001*. August 2001, Computer Graphics Proceedings, Annual Conference Series, pp. 519–528, ACM Press / ACM SIGGRAPH.
- [29] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, and Westley Sarokin, “Acquiring the reflectance field of a human face,” in *Computer Graphics Proceedings, Annual Conference Series, SIGGRAPH 2000*, July 2000, pp. 145–156.
- [30] Daniel Lischinski, Filippo Tampieri, and Donald P. Greenberg, “Discontinuity meshing for accurate radiosity,” *IEEE Computer Graphics and Applications*, vol. 12, no. 6, pp. 25–39, Nov. 1992.
- [31] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, Interscience Publisher, Inc., New York, 1953.
- [32] James T. Kajiya, “Anisotropic reflection models,” in *Computer Graphics (SIGGRAPH ’85 Proceedings)*, July 1985, vol. 19, pp. 15–21.
- [33] Brian Cabral, Nelson Max, and Rebecca Springmeyer, “Bidirectional reflection functions from surface bump maps,” in *Computer Graphics (SIGGRAPH ’87 Proceedings)*, July 1987, vol. 21, pp. 273–281.
- [34] Francois X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg, “A global illumination solution for general reflectance distributions,” in *Computer Graphics (SIGGRAPH ’91 Proceedings)*, July 1991, vol. 25, pp. 187–196.

- [35] Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng, “Sampling with hammersley and halton points,” *Journal of Graphics Tools*, vol. 2, no. 2, pp. 9–24, 1997.
- [36] Jianjun Cui and Willi Freeden, “Equidistribution on the sphere,” *SIAM Journal on Scientific Computing*, vol. 18, no. 2, pp. 595–609, 1997.
- [37] R. L. Joshi, H. Jafarkani, J. H. Kasner, T. R. Fischer, N. Farvardin, M. W. Marcellin, and R. H. Bamberger, “Comparison of different methods of classification in subband coding of images,” *IEEE Transaction on Image Processing*, vol. 6, pp. 1473–1486, 1997.
- [38] K. R. Rao, *Discrete Cosine Transform, Algorithms, Advantages and Applications*, Academic Press, 1990.
- [39] R. M. Gray, *Source Coding Theory*, Kluwer Academic, 1990.
- [40] R.C. Reininger and J.D. Gibson, “Distribution of the two-dimensional dct coefficients for images,” *IEEE Transactions on Communications*, vol. COM-31, pp. 835–839, 1983.
- [41] A. N. Akansu and M. J. T. Smith, *Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets*, Academic Press, 1992.
- [42] Roy Hall, *Illumination and Color in Computer Generated Imagery*, Springer-Verlag, New York, 1989.
- [43] M. Charriera, D. S. Cruz, and M. Larsson, “JPEG2000, the next millenium compression standard for still images,” in *Proceedings of the IEEE ICMCS’99*, June 1999.
- [44] Peter Schröder and Wim Sweldens, “Spherical wavelets: Efficiently representing functions on the sphere,” in *SIGGRAPH 95 Conference Proceedings*, Robert Cook, Ed. ACM SIGGRAPH, Aug. 1995, Annual Conference Series, pp. 161–172, Addison Wesley, held in Los Angeles, California, 06-11 August 1995.
- [45] Eric P. F. Lafourture, Sing-Choong Foo, Kenneth E. Torrance, and Donald P. Greenberg, “Non-linear approximation of reflectance functions,” in *Computer Graphics Proceedings, Annual Conference Series 1997*. August 1997, pp. 117–126, ACM SIGGRAPH.
- [46] A. Said and W. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–249, 1996.
- [47] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.