Illuminating Image-based Objects

Tien-Tsin Wong[1] ttwong@acm.org

Pheng-Ann Heng[1] pheng@cse.cuhk.edu.hk

Siu-Hang Or[1] shor@cse.cuhk.edu.hk

Wai-Yin Ng[2] wyng@ie.cuhk.edu.hk

[1]Department of Computer Science & Engineering
[2]Department of Information Engineering
The Chinese University of Hong Kong

Abstract

In this paper, we present a new scheme of data representation for image-based objects. It allows the illumination to be changed interactively without knowing any geometrical information (e.g. depth or surface normal) of the scene. But the resulting images are physically correct. The scene is first sampled from different view points and under different illuminations. By treating each pixel on the image plane as a surface element, *the sampled images are used to measure the* apparent *BRDF of each* surface element. *Two compression schemes,* spherical harmonics *and* discrete cosine transform, *are proposed to compress the tabular BRDF data. Whenever the user changes the illumination, a certain number of views are reconstructed. The correct user perspective view is then displayed using the standard texture mapping hardware. Hence, the intensity, the type and the number of the light sources can be manipulated interactively.*

CR Categories: I.3.2[Computer Graphics]: Picture/Image Generation — Digitizing and scanning, viewing algorithms.
Additional keywords: image-based rendering, spherical harmonics, light field, Lumigraph, holographic stereogram, BRDF, illumination.

1. Introduction

Although millions of textured micropolygons can be rendered within a second using state-of-the-art graphics workstations, rendering a realistic complex scene at interactive speed is still difficult. Unlimited complexity of the scene and expensive modeling cost are two major problems. Recently researchers have focused on a new approach to rendering, namely, image-based rendering. This approach breaks the dependency of rendering time on the scene complexity, since the rendering primitives are no longer geometrical entities, but images.

Previous work can be classified into two main streams. The first stream focuses on determining the correct perspective view. Foley *et al.* [8] developed a system which can rotate raytraced voxel data interactively by view interpolation. However, their interpolation method is not physically correct. Chen and Williams [6] interpolated views by modeling pixel movement, resulting in physically correct interpolation. Later, Chen [4] described an image-based rendering system, QuickTime VR, which generates perspective views from panoramic image data by warping [5]. McMillan and Bishop [14] mentioned that image-based rendering is a problem of finding and manipulating the plenoptic function [1]. They also proposed a method to sample and reconstruct this plenoptic function. Levoy and Hanrahan [13] and Gortler *et al.* [9] reduced the 5D plenoptic function to a 4D light field or Lumigraph. This simplification allows the view interpolation to be done by standard texture mapping techniques, which can be further accelerated by hardware. Recently, animated image-based objects are developed by Live Picture [16].

The second stream of research focuses on re-rendering the scene under different illumination using the sampled images. Haeberli [10] re-rendered the scene using simple superposition property. However, the direction, the type and the number of the light sources are limited to the lighting setup during capturing the scene. Nimeroff *et al.* [15] efficiently re-rendered the scene under various natural illumination (overcast or clear skylight) with the knowledge of the empirical formulæ that model the skylight. Belhumeur and Kriegman [2] determined the basis images of an object with the assumptions that the object is convex, and all surfaces are Lambertian. With these assumptions, only three basis images are enough to span the *illumination cone* of the object, *i.e.*, three images are enough to reconstruct/re-render the scene under various illuminations.

In the first stream of previous work, the illumination of the scene was assumed to be fixed and carefully designed. On the other hand, the view point is assumed fixed in the work of second stream. In this paper, we present an image-based rendering system which allows the change of view point as well as the change of illumination. All image-based objects can be described as a special form of plenoptic function. Most previous work assumed that the time parameter t of the plenoptic function was fixed. The method described in this paper can be thought of as an attempt to allow t to vary.

There are two major motivations for this research. Firstly, the variability of the illumination allows the viewer to illuminate only interesting portions of the scene. This improves the viewer's recognition of it. Secondly, it is a step closer to realizing the use of image-based entities (plenoptic function, light field or Lumigraph) as basic rendering entities, just like geometrical entities used currently in conventional graphics systems.

One major goal of image-based rendering is to minimize the use of geometrical information while generating physically correct images. With this goal in mind, the proposed image-based system allows the viewer to change the scene lighting interactively without knowing geometrical details (say, depth or surface normal) of the scene. The *apparent* BRDF [12, 19] of each pixel on the image plane is sampled. With these pixel BRDFs, physically correct views of the scene can be reconstructed under different illuminations by fitting different lighting parameter values and viewing direction. The BRDF data representation is described in Section 2. Section 3 describes how the light source can be manipulated once the pixel BRDFs are recorded. Two compression schemes, spherical harmonic transform and discrete cosine transform, are proposed to compress the huge amount of BRDF data. They are discussed in Section 4.

The proposed BRDF representation is general enough to be applied to a wide range of image-based objects, including panoramic image, plenoptic function, light field and Lumigraphs. In this paper, we demonstrate how to extend light field and Lumigraph systems in order to allow illumination changes. The reason to choose light field and Lumigraphs is due to their simplicity and potential to utilize graphics hardware. But this does not imply the proposed BRDF representation is only valid for light slab based objects. Further discussions and conclusions on the new data representation can be found in Sections 5 and 6 respectively.

2. BRDF Representation

2.1. BRDF of Pixel

The *bidirectional reflectance distribution function* (BRDF) [12] is the most general form of representing surface reflectivity. To calculate the radiance outgoing from a surface element in a specific direction, the BRDF of this surface element must first be determined. Methods for measuring and modeling the BRDF can be found in various sources [3, 19]. The most straightforward approach to include the illumination variability of the image-based rendering system is to measure the BRDF of each object material visible in the image. However, this approach has several drawbacks. While the BRDFs of synthesized object surfaces may be assigned at will, measuring those of all objects in a real scene is tedious and often infeasible. Imagine a scene containing thousands of small stones, each with its own BRDF. The situation worsens when a single object exhibits spatial variability of surface properties. Furthermore, associating an BRDF to each object in the scene causes rendering time to depend on the scene complexity.

One might suggest, for each pixel in each view, to measure the BRDF of the object surface seen through that pixel window. This approach breaks the link to the scene complexity, but introduces an aliasing problem. Consider pixel A in Figure 1: multiple objects are visible through the pixel window. Note that this will frequently happen in images showing distant objects. Even if only one object is visible, there is still the problem of choosing surface normal for measuring BRDF when the object silhouette is curved (see pixel B in Figure 1).

Our solution is to treat each *pixel* on the image plane as a surface element with an *apparent* BRDF. Imagine the image plane as just a normal planar surface, and each pixel can be regarded as a surface element. Each surface element emits different amounts of radiant energy in different directions under different illuminations. In order to measure the (apparent) BRDF of each pixel, the location of the image plane must be specified (see Figure 2), not just the direction. By recording the BRDF of each pixel (Figure 2), we capture the aggregate reflectance of objects visible through that pixel window. The light vector L from the light source and the viewing vector V from the view point E define the

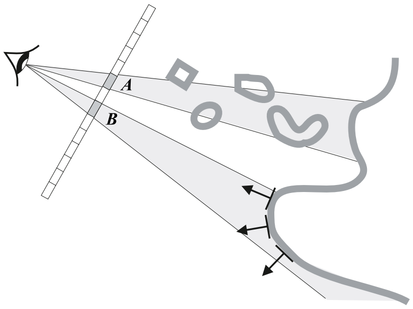


Figure 1. Aliasing problem of measuring object surface visible through the pixel windows.

two directions of the BRDF. This approach does not depend on the scene complexity, and removes the aliasing problems above. Moreover, it can be easily integrated in the light slab based data structure [13, 9]. It is also a unified approach for both virtual and real world scenes.

Note that the apparent BRDF represents the response of the object(s) in a pixel to light in each direction, *in the presence of the rest of the scene*, not merely the surface reflectivity. If we work from views (natural or rendered) that include shadows, therefore, shadows appear in the reconstruction.

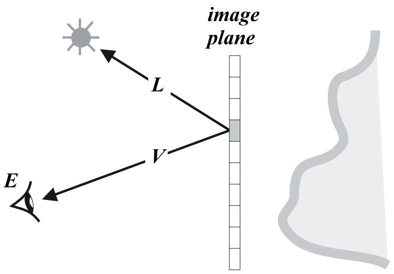


Figure 2. Measuring the BRDF of the *pixel*.

2.2. Measuring BRDF

To measure the BRDF, we have to capture the image of the virtual or real world scene under different illuminations. A directional light source is cast on the scene from different directions. Rendered images and photos of the virtual or real world scene are captured as usual. The algorithm is,

```
For each view point $E$
     For each directional light source's direction $(\theta, \phi)$
          Render the virtual scene or take photograph of
          real world scene illuminated by this
          directional light source and named as $I_{E,\theta,\phi}$.
```

The parameter θ is the polar angle, and ϕ is the azimuth. The direction $(0, \phi)$ is orthogonal to the image plane. The parameters are localized to the image plane coordinate system, so transforming the image plane does not affect the BRDF parameterization. The reason for using a directional light source is that the incident light direction is identical at any 3D point. In real life, a directional light source can be approximated by placing a spotlight at a sufficient distance from the scene. The BRDF ρ of each pixel inside a view can be sampled by the following algorithm,

```
For each view point $E$
     For each pixel $(s, t)$
          $\rho(\theta, \phi) = \frac{\text{pixel value at } (s,t) \text{ of } I_{E,\theta,\phi}}{\text{intensity of light source}}$
```

One assumption is that there is no intervening medium, which absorbs, scatters or emits any radiant energy.

Note that instead of recording a single 2D array (image plane) of pixel BRDFs, we record a set of 2D arrays of pixel URDFs (described shortly) from multiple view points (E) in our current implementation. Since the viewing direction of each pixel within one specific view of the image plane is fixed, the BRDF ρ is simplified to a unidirectional reflectance distribution function (URDF) which depends on the light vector only. Hence, the function ρ is parameterized by two parameters (θ, ϕ) only. There are three reasons we store the partial URDF of each pixel in multiple fixed views, instead of a complete BRDF for each pixel of a single image plane. Firstly, with this organization, the compression methods (described in Section 4) are simplified. Secondly, the reconstruction from compressed data is performed only when the lighting changes. No reconstruction is needed when the user changes view point. This is important for interactive display, since the user changes the view point more often than the illumination. Thirdly, pixels on the same image do not have the same viewing vector V. Resampling is needed to sample the complete pixel BRDF on the uniform spherical grid. Hence this organization free us from the resampling process which may introduce error. From now on, the term *view* refers to an image of the image plane, viewed from certain view point (E). When we refer *BRDF*, we actually means the set of partial URDFs in multiple fixed views.

Traditionally, the BRDF is sampled only on the upper hemisphere of the surface element, since reflectance must be zero if the light source is behind the surface element. However in our case, the reflectance may be nonzero even the light source direction is from the back of the image plane. This is because the actual object surface may not align with the image plane (Figure 3). Instead, the whole sphere surrounding the pixel has to be sampled for recording its BRDF. Therefore, the range of θ should be $[0, \pi]$. Nevertheless, sampling only the upper hemispherical BRDF is usually sufficient, since the viewer seldom moves the light source to the back of objects.

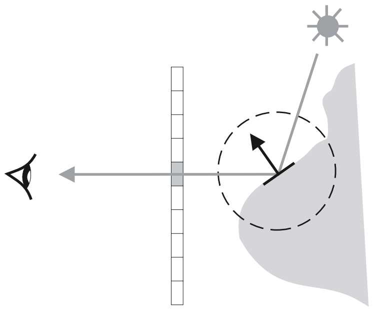


Figure 3. The image plane may not be parallel with the object surface.

3. Manipulating the Light Sources

Once the BRDFs are sampled and stored, they can be manipulated. The final radiance (or simply value) of each pixel in each view is determined by evaluating equation 1, given the intensity and the direction of the light sources.

$$\text{value at pixel } (s,t) \text{ in a view } (E) = \sum_{i}^{n} \rho_{E,s,t}(\theta_i,\phi_i) I_i, \tag{1}$$

where n is the total number of light sources,
(θ_i,ϕ_i) specify the direction of the i-th light source L_i,
I_i is the intensity of the i-th light source.

Note this equation will give a physically correct image. This can be proved with as follows. Consider k unoccluded objects, visible through the pixel (s,t), viewed from view point E and illuminated by n light sources. The radiance passing through the pixel window in this view will be,

$$\begin{aligned} & \sum_{i}^{n} \rho_i^0 I_i + \sum_{i}^{n} \rho_i^1 I_i + \cdots + \sum_{i}^{n} \rho_i^k I_i \\ = & \sum_{j}^{k} \rho_0^j I_0 + \sum_{j}^{k} \rho_1^j I_1 + \cdots + \sum_{j}^{k} \rho_n^j I_n \\ = & \rho_0 I_0 + \rho_1 I_1 + \cdots + \rho_n I_n \end{aligned}$$

where ρ_i^j is the reflectance of the j-th object illuminated by light L_i,
$\rho_i = \sum_{j=1}^{k} \rho_i^j$ is the aggregate reflectance we recorded when measuring the BRDF of the pixel.

Light Direction With equation 1, the light direction can be changed by substituting a different value of (θ,ϕ). Figures 12(a) and (b) show a teapot illuminated by a light source from the top and the right respectively.

Light Intensity Another parameter to manipulate is the intensity of the light source. This can be done by changing the value of I_i for the i-th light source. Figure 13(a) shows the Beethoven statue illuminated by a blue light from the left.

Multiple Light Sources We can arbitrarily add any number of light sources. The trade-off is the computational time. From equation 1, a new reflectance ρ_i has to be reconstructed from compressed data (described in Section 4) for each light source. Our current prototype can still run at an acceptable interactive speed using up to 3 directional light sources. In the Figure 13(b), the Beethoven statue is illuminated by a blue light from the left and a red light from the right simultaneously.

Type of light sources Up to now, we have made an implicit assumption that the light source for manipulation is directional. Directional light is very efficient in evaluating equation 1, because all pixels on the same image plane are illuminated by light source from the same direction (θ_i,ϕ_i). However, the method is not restricted to directional light. It can be extended to point source and spotlight also. However, it will be more expensive to evaluate equation 1 for other type of light sources, since (θ_i,ϕ_i) will need to be recalculated from pixel to pixel.

Since the image plane where the pixels are located is only a window in the 3D space (Figure 2), the intersected surface element that actually reflects the light may be located on any point on the ray V in Figure 4. To find the light vector L correctly for other types of light sources, the intersection point of the ray and the object surface have to be located first. Note there is no such problem for directional source, since the light vector is same for all points in the 3D space. One way to find L is to use the depth image. While this can be easily done for rendered images, real world scenes may be more difficult. Use of a range scanner may provide a solution. Figures 14(a) and (b) show a box on a plane illuminated by a point source and a directional source respectively. Note the difference in the shadow cast by these sources. However, just as we discussed in Section 2, there is an aliasing problem in finding the correct positions of intersecting objects. Imagine a scene of a furry teddy bear; thousands of objects may be visible through one pixel window.

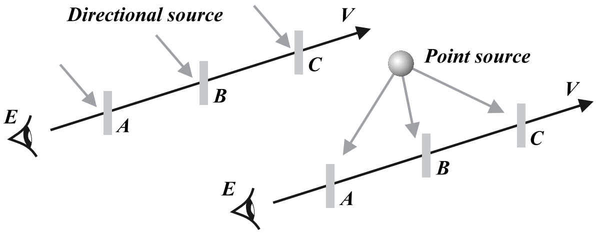


Figure 4. Finding the correct light vector.

4. Compression

Storing the whole BRDF tables requires an enormous storage space. For a single pixel, if the URDF is sampled in the polar coordinate system with 20 samples along both the θ and ϕ coordinates, there will be 400 floating point numbers stored for each pixel. A single view of a 256×256 image plane will require 100Mb of storage.

To represent the BRDF more efficiently, the tabular data is transformed to the frequency domain and quantization is performed to reduce storage. We have tested two types of transforms, spherical harmonic transform and discrete cosine transform.

4.1. Spherical Harmonics

Spherical harmonics [7] are analogous to a Fourier series in the spherical domain. Cabral *et al.* [3] proposed the representation of BRDF using spherical harmonics. The work is further extended by Sillion *et al.* [18] to model the entire range of incident angle. It is especially suitable for representing smooth spherical functions. In our approach, the viewing direction V for each pixel is actually fixed. Hence, the function ρ can be transformed to spherical harmonics domain using the following equations directly, without considering how to represent a bidirectional function described as in [18].

$$C_{l,m} = \int_0^{2\pi} \int_0^{\pi} \rho(\theta, \phi) Y_{l,m}(\theta, \phi) \sin\theta d\theta d\phi,$$

where

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

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

and

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

where the base case is $P_{0,0}(x) = 1$.

$C_{l,m}$'s are the coefficients of the spherical harmonics which are going to be stored for each pixel. The more coefficients are used, the more accurate the spherical harmonics representation is. Accuracy also depends on the number of samples in the (θ, ϕ) space. We found 16 to 25 coefficients are sufficient in most of our tested examples.

To reconstruct the reflectance given the light vector (θ, ϕ), the following equation is solved for each pixel in each view.

$$\rho(\theta,\phi) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} C_{l,m} Y_{l,m}(\theta,\phi). \qquad (2)$$

where $(l_{max})^2$ is the number of spherical harmonics coefficients to be used.

Figure 5 shows the sampled reflectance distribution of a pixel on the left and its corresponding reconstructed distribution on the right. There are 900 samples (30 along θ in the range $[0, \frac{\pi}{2}]$ and 60 along ϕ) in the left original distribution. The reconstructed distribution on the right is represented by 25 spherical harmonics coefficients only.

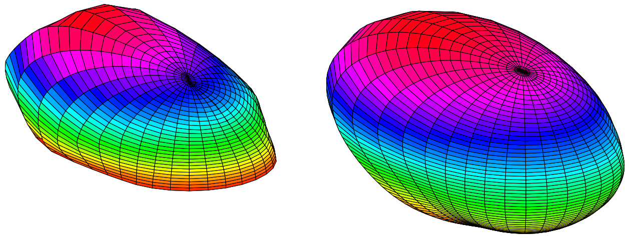


Figure 5. Original sampled (left) and reconstructed (right) distribution. Note the lower hemisphere of the reconstructed distribution is interpolated to prevent discontinuity.

4.2. Discrete Cosine Transform

Although spherical harmonics can efficiently represent smooth spherical functions, it is inferior to represent discontinuous function which is quite common if the scene contains shadow. This phenomenon motivates us to find another solution for data compression.

The second compression scheme we have tested is discrete cosine transform (DCT). One reason to choose DCT is that hardware DCT codec is becoming widely available. Same as before, we do not compress the four dimensional BRDFs. Instead, the two dimensional URDFs are compressed. Since the URDF is a spherical function, it is first mapped to a 2D disc (Figure 6), before applying the standard 2D discrete cosine transform to the disc image.

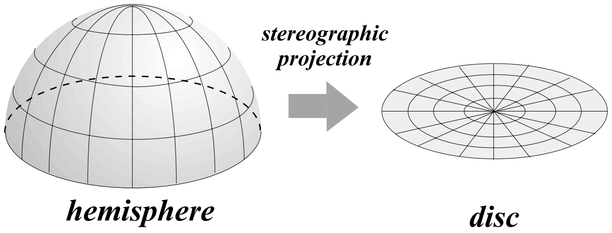


Figure 6. Mapping a hemisphere to a disc.

To map a spherical function to a plane, the mapping should be done in two passes, namely, one for the upper hemisphere and one for the lower half. The mapping from a hemisphere to a disc is done by stereographic projection [11]. To project the lower hemisphere, the point of projection C is first placed at the pole of upper hemisphere and the plane is placed underneath the lower hemisphere (Figure 7). A point S on the hemisphere is mapped to point P on the plane by firing a ray from C through point S and intersects the plane at point P. The upper hemisphere can be mapped to plane similarly. The polar coordinate (θ, ϕ) on a hemisphere is mapped to the 2D coordinate (x, y), within a unit square by

$$\begin{aligned} x &= \frac{1}{2}[\tan(\theta/2)\cos(-\phi) + 1] \\ y &= \frac{1}{2}[\tan(\theta/2)\sin(-\phi) + 1] \end{aligned}$$

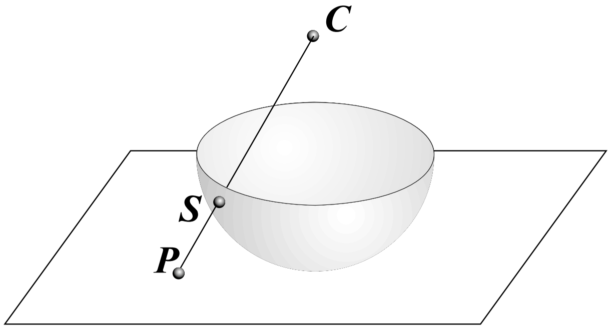


Figure 7. Stereographic projection.

The resultant disc image after projecting the upper hemisphere of the URDF of an example pixel is shown in Figure 8(a). The example pixel is extracted from the test scene in Figure 9(a). In Figure 8(a), the white region near the image center indicates the specular highlight. The polygonal black hole on the left is due to the shadow cast by the box in Figure 9(a).

Once the spherical function is projected to 2D image, discrete cosine transform (DCT) can be applied to transform the image and the resulting DCT coefficients are zonal sampled and quantized. Figure 8 shows the images before (a) and after (b) the quantization in DCT domain. Only 64 coefficients are retained for the image in Figure 8(b), while the image in Figure 8(a) is represented by 50×50 floating point data.

4.3. Visual Comparison

Figure 9 visually compares the reconstructed images of different compression schemes. Figure 9(a) shows the test scene containing a box which cast shadow on a plane. The square region in Figure 9(a) is enlarged for visual comparison. The ideal result is generated by looking up the uncompressed BRDFs (Figure 9(b)). Note that the hard shadow is preserved. After the stereographic projection, some errors are introduced. It is because the mapping process is actually a resampling process. Blurring is found around the shadow in Figure 9(c). This error can be reduced by increasing the resolution of the disc image, *i.e.* increase the number of samples. However, the storage size will also be increased. Figure 9(d) shows the reconstructed image generated from data compressed using spherical harmonics. The error in this image is purely due to the quantization taken place in the spherical harmonic domain. Figure 9(e) shows the reconstructed image generated from data compressed using DCT. The error in this image includes quantization error in frequency domain and the resampling error during stereographic projection.

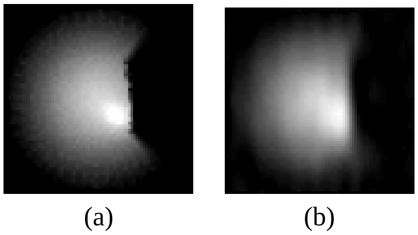
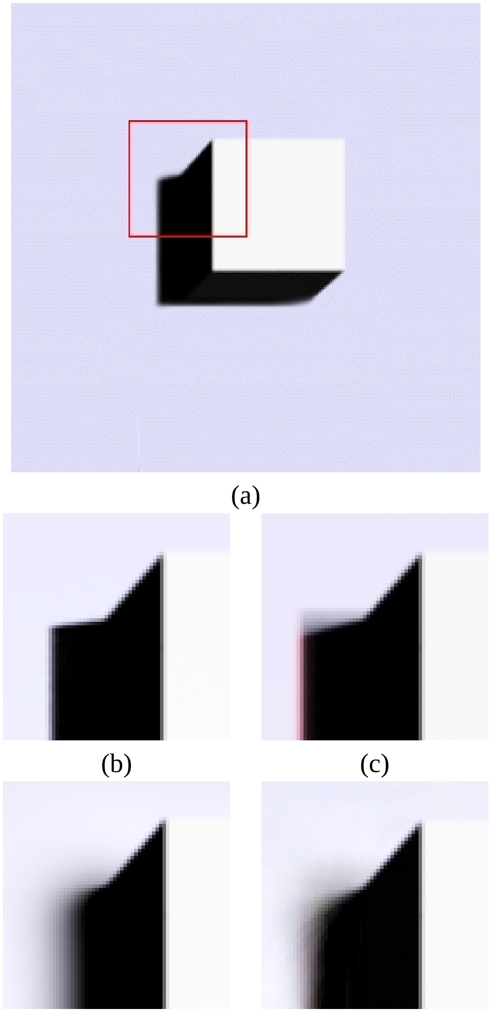


Figure 8. Before (a) and after (b) quantizing the disc image in frequency domain. Original data in (a) is represented by 50 x 50 floating point data. The number of coefficients to represent the image (b) is 64.

In order to have a fair comparison, equal number of coefficients (64 floating point coefficients) are used to compress the data in both compressed cases (Figures 9(d) & (e)). Comparing Figure 9(d) to Figure 9(e), the image generated from DCT is more noisy than that of spherical harmonics. However, the shadow in the image generated from DCT is a better approximation of the true shadow in Figure 9(b). The sharp corner of the shadow becomes a round corner in the case of spherical harmonics, while the corner is still observable in the case of DCT. This is also confirmed by the RMS of error statistics. The RMS of error of image generated from spherical harmonic data is 0.1043 while that of image generated from DCT data is 0.0865. From this experiment, DCT compression scheme is preferred if the scene contains hard shadows and a close approximation to the true image is needed. On the other hand, spherical harmonics is preferred if the scene contains not much hard shadow and a pleasant (smooth) visual result is a main concern.

5 Discussion

Extending Light Field and Lumigraph We have developed an image-based rendering system based on the light field [13] and Lumigraph systems [9], due to their simplicity and efficiency. The proposed idea is tested in this light slab



(a)

(b) (c)

(d) (e)

Figure 9. Visual comparison of reconstructed images. (a) Test scene. (b) Image generated using uncompressed BRDFs. (c) Result after projecting spherical function to a disc, also uncompressed. RMS(E)=0.0979. (d) Result from data compressed using spherical harmonics. RMS(E)=0.1043. (e) Result from data compressed using DCT. RMS(E)=0.0865.

based system. No modification to our original BRDF representation is needed. Light slab based system represents image-based object by a two-plane data structure. Following the terminology of Levoy and Hanrahan [13], the front and back planes of the light slab are denoted as uv and st planes. Both planes are parameterized by grids (uv and st grids). The viewing vector V is implicitly specified by the light slab parameters (u, v, s, t) (see Figure 10). The viewpoint E is replaced by a point (u, v) on the uv grid of the front plane of the light slab. The measurement of pixel BRDFs is done as before.

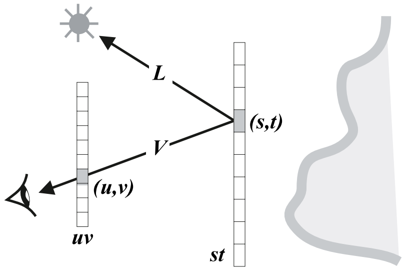


Figure 10. Extending the light slab based systems to allow change of illumination.

Implementation We have implemented the method described, and developed an interactive viewer with controllable illumination. We follow the texture mapping approach described by Gortler *et al*. [9] to display the light slab. With hardware texture mapping, the scene can be rotated, panned and zoomed at an interactive speed on a SGI Indigo 2 with High Impact. The current version only supports the spherical harmonics compression scheme. The reconstruction of views is performed whenever the user changes the light source direction and it is done purely by software. Nevertheless, the program can still update the image at an interactive speed when the viewer drags the light sources around.

Further Compression For a view (image of st plane) with a resolution of 256×256, where each channel (R,G,B) of a pixel BRDF is represented by 25 floating point coefficients, 18.75 Mb of storage are required. By storing a bitmap indicating which coefficient vectors are nonzero and storing only those nonzero vectors, the necessary storage usually drops to $2 \sim 3$ Mb. Another way to compress the data is to use a variable length coefficient vector, since not all pixel BRDFs need the same number of coefficients to represent. We have not yet further compressed the transformed and quantized data by standard entropy coding, such as LZW or Huffman code. More compact data size may be resulted if the entropy coding is done. Moreover, we believe vector quantization is promising for further compression of the coefficient vectors, since it explores the coherence between adjacent pixels.

Preventing Discontinuity While smooth functions can be transformed to frequency domain and represented by finite number of coefficients, discontinuous functions require an infinite number of coefficients to represent. Truncating the spherical harmonic series or discrete cosine series gives

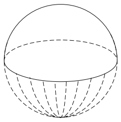


Figure 11. The boundary value along the equator is linearly interpolated to prevent equatorial discontinuity in the sampled BRDF.

persistent Gibb's ringing artifacts. One source of discontinuity is the incomplete sampling of light directions (boundary discontinuity). Incomplete sampling is sometimes necessary for fast scene sampling. From our experience, there is no need to sample the whole range of θ, *i.e.*, $[0, \pi]$. Usually the range $[0, \frac{\pi}{2}]$ is sufficient. Zeroing all the unsampled entries introduces discontinuity along the equator of the sampling sphere. To avoid this sharp change, the boundary value along the equator is linearly interpolated to a constant value at the south pole (see Figure 11 and the right diagram in Figure 5). Another source of discontinuity is shadowing (Figure 8(a)), which is unavoidable. Hard shadows will be smoothed out if represented by a finite sum of harmonics (Figure 9(d) & (e)).

Independence of Sampled uv Resolution Whenever the viewer changes the lighting, a set of images viewed from some coordinates (u, v) on the uv plane is reconstructed for display. In our implementation, the images are reconstructed at the sampled (u, v) coordinates. However, one interesting result of recording pixel BRDFs instead of images is that images for (u, v) positions other than the sampled locations may be generated by resampling. Moreover, the images can be reconstructed at a different resolution of the uv grid. This implies that we can sample the scene at a high resolution of the uv grid but reconstruct the views at a different uv resolution, depending on the capability of the hardware. This is quite different from the previous light slab based systems [13, 9] which represents the scene by storing a set of images viewed from certain view points.

6 Conclusions and Future Work

We have proposed and implemented a new method to allow the image-based objects to be displayed under varying illumination. It is especially efficient when illuminated by directional light sources. The use of apparent pixel BRDFs instead of the image set allows reconstruction, and hence display of the scene at different uv resolutions. This is especially useful when the image-based object is rendered on machines with a lower rendering capability.

Spherical harmonics and discrete cosine transforms are tested and compared. It is shown that spherical harmonic transform is inferior to represent scene containing hard shadow but gives pleasant images. Discrete cosine transform gives noisy images but gives better approximation to the true solution. Another potential compression scheme is spherical wavelets [17]. Belhumeur and Kriegman's model [2] is another efficient approach in representing convex diffuse objects. Its application to represent general objects requires further investigation.

Currently, all of our tested data are synthetic scenes. We are undertaking the capture of real world scenes with a hand held camera.

There is still much work to do in using the image-based object as a basic rendering primitive, and our work is only a preliminary step in this direction.

Acknowledgements

We would like to thank all the reviewers for their valuable suggestions and comments.

Web Availability

All latest supplementary materials are available on the web page at:
`http://www.cse.cuhk.edu.hk/~ttwong/`

References

[1] E. H. Adelson and J. R. Bergen. *The Plenoptic Function and the Elements of Early Vision*, chapter 1. The MIT Press, Cambridge, Mass, 1991.

[2] Peter N. Belhumeur and David J. Kriegman. What is the set of images of an object under all possible lighting conditions? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1996.

[3] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 273–281, July 1987.

[4] Shenchang Eric Chen. QuickTime VR - an image-based approach to virtual environment navigation. In Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'95*, pages 29–38, August 1995.

[5] Shenchang Eric Chen and G. S. P. Miller. Cylindrical to planar image mapping using scanline coherence. United States Patent number 5,396,583, March 7 1995.

[6] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, 1993.

[7] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. Interscience Publisher, Inc., New York, 1953.

[8] Thomas A. Foley, David A. Lane, and Gregory M. Nielson. Towards animating raytraced volume visualization. *The Journal of Visualization and Computer Animation*, 1(1):2–8, 1990.

[9] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In Computer Graphics *Proceedings*, *Annual Conference Series, SIGGRAPH'96*, pages 43–54, August 1996.

[10] Paul Haeberli. Synthetic lighting for photography. Published on the web page GRAFICA Obscura at `http://www.sgi.com/grafica/synth/index.html`, January 1992.

[11] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination*. Chelsea Publising Company, New York, 1952.

[12] James T. Kajiya. Anisotropic reflection models. In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 15–21, July 1985.

[13] Marc Levoy and Pat Hanrahan. Light field rendering. In Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'96*, pages 31–42, August 1996.

[14] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'95*, pages 39–46, August 1995.

[15] Jeffry S. Nimeroff, Eero Simoncelli, and Julie Dorsey. Efficient re-rendering of naturally illuminated environments. In *Fifth Eurographics Worksop on Rendering*, pages 359–373, Darmstadt, Germany, June 1994.

[16] Live Picture. Realspace viewer 1.0, 1997. A software to view image-based objects.

[17] Peter Schroder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 161–172, August 1995.

[18] 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)*, volume 25, pages 187–196, July 1991.

[19] Gregory J. Ward. Measuring and modeling anisotropic reflection. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.

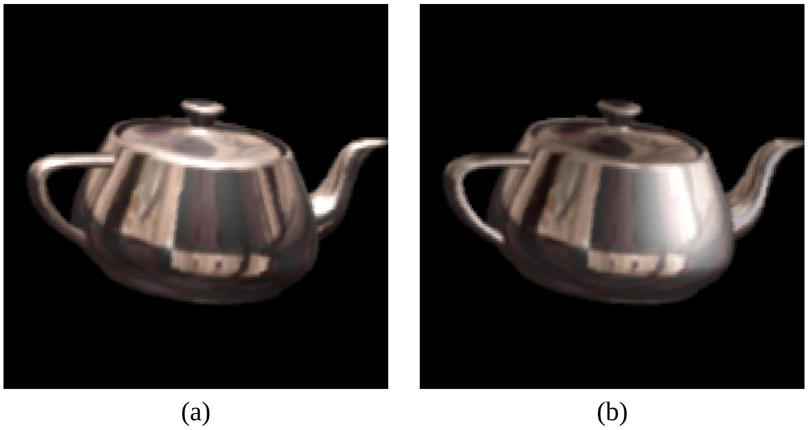


Figure 12. (a)Left: light from the top of the teapot. (b)Right: light from the right side. No. of spherical harmonics (s.h.): 25, st resolution (res.): 256x256.

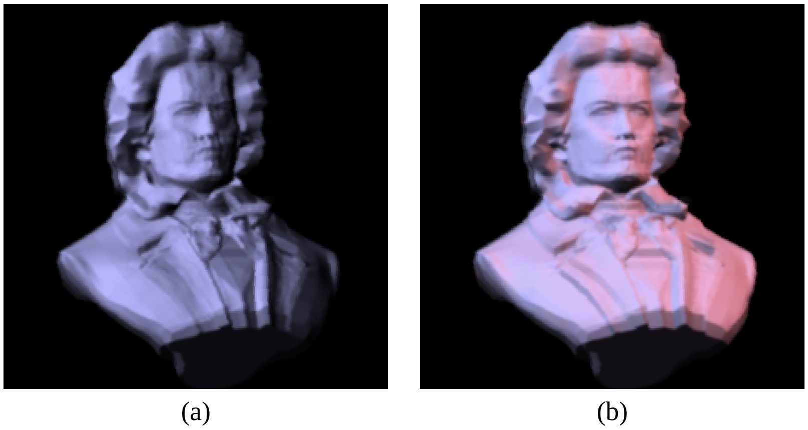


Figure 13. (a)Left: Beethoven statue illuminated by a single blue light from the left. (b)Right: One more red light comes from the right. No. of s.h.: 25, st res.: 256x256.

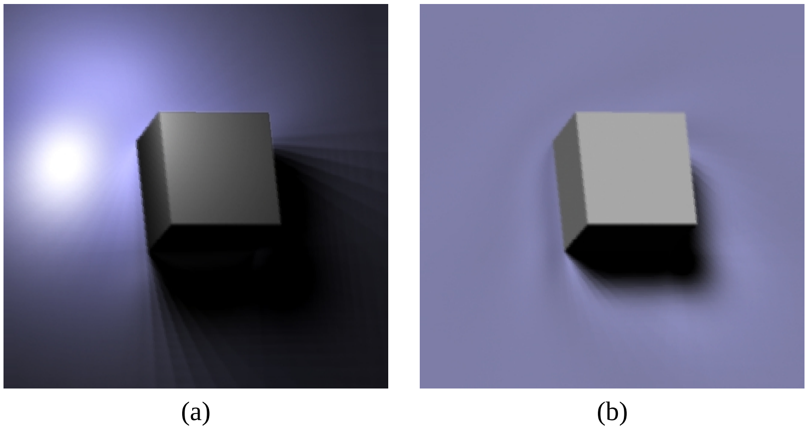


Figure 14. a)Left: shadow cast by a point source. (b)Right: shadow cast by a directional source. No. of s.h.: 64, st res.: 256x256.