Image-based Rendering with Controllable Illumination

Tien-Tsin Wong[1] Pheng-Ann Heng[1] Siu-Hang Or[1] Wai-Yin Ng[2]

[1] Department of Computer Science & Engineering,
[2] Department of Information Engineering,

The Chinese University of Hong Kong

Abstract: A new image-based rendering method, based on the light field and Lumigraph system, allows illumination to be changed interactively. It does not try to recover or use any geometrical information (*e.g.*, depth or surface normals) to calculate the illumination, but the resulting images are physically correct. The scene is first sampled from different viewpoints and under different illuminations. Treating each pixel on the back plane of the light slab as a *surface element*, the sampled images are used to find an *apparent BRDF* of each *surface element*. The tabular BRDF data of each pixel is further transformed to the spherical harmonic domain for efficient storage. Whenever the user changes the illumination setting, a certain number of views are reconstructed. The correct user perspective view is then displayed using the texture mapping technique of the Lumigraph system. Hence, the intensity, the type and the number of the light sources can be manipulated interactively.

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 scene complexity and high modeling cost are major problems. Recently, research has focused on a new approach; image-based rendering, which breaks the dependency of rendering time on the scene complexity. The rendering primitives are no longer geometrical entities, but sampled images.

1.1 Previous works

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 a panoramic image by warping [5]. McMillan and Bishop [12] mentioned that image-based rendering is a problem of finding and manipulating the plenoptic function [1], and proposed a method to sample and reconstruct it. Levoy and Hanrahan [11] 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 in turn be further accelerated by hardware.

Nimeroff *et al.* [13] efficiently re-rendered the scene under various natural illumination (overcast or clear skylight) by linearly combining a set of pre-rendered basis images. 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.

1.2 Overview

Previous research mainly concentrated on finding the correct perspective view interpolation. The illumination of the scene was assumed fixed and carefully designed. In this paper, we present an image-based rendering system which allows variable illumination. The illumination is not restricted to specific form as in the approach of Nimeroff *et al.* [13]. Moreover, there is no restriction on the shape of the object nor the type of surfaces, as in the model of Belhumeur and Kriegman [2].

This system is based on the light field and Lumigraph systems. However, it can also apply to a general plenoptic rendering system. Previous work assumed that the time parameter t of the plenoptic function is fixed. Our method can be thought of as an attempt to allow t to vary.

There are two major motivations for this research. Firstly, the variability allows the viewer to illuminate only interesting portions of the scene, improving recognition of it. Secondly, it is a step closer to realizing the use of image-based entities (plenoptic function, light field or Lumigraph) as the basic rendering entities, just as 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 in mind, we have developed an image-based system which allows the viewer to change the scene lighting interactively without knowing the geometrical details (say, depth or surface normal) of the scene. The BRDF [10, 15] of each pixel within each view is first sampled and stored as a set of spherical harmonic coefficients. With these pixel BRDFs, physically correct views of the scene can be reconstructed under different illuminations. Storing pixel BRDFs allows us to represent the radiance in various viewing directions as a set of functions (BRDFs), instead of a set of fixed views used in previous light field rendering and Lumigraph systems. This provides the potential for further compression. It also allows the uv resolution during rendering (defined in section 2) to be independent of the sampled uv resolution, so that the scene can be rendered at a lower uv resolution by limited hardware, independent of the resolution of the original sample.

2 Image-based Rendering with Controllable Illumination

Our image-based rendering system is based on the light field [11] and Lumigraph systems [9], due to their simplicity and efficiency. We follow the terminology used by Levoy and Hanrahan [11], and call the data structure for image representation the *light slab*. The front and back planes of the light slab are denoted as uv and st planes respectively. The terms *view* and *image* are used interchangeably. A *view* is an image of the st plane, viewed from a particular (u, v) coordinate on the uv plane. Our work mainly focuses on the representation of image data and the reconstruction of correct views for the purpose of interactive navigation.

2.1 BRDF Representation

The *bidirectional reflectance distribution function* (BRDF) [10] 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, 15]. 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).

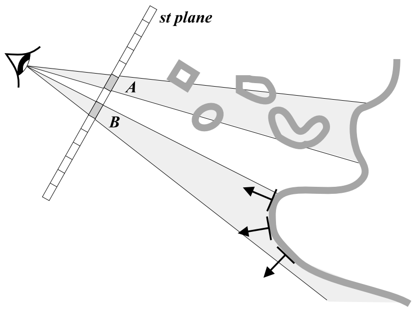


Fig. 1. Aliasing problem of measuring object surface visible through the pixel windows.

Our solution is to treat each *pixel* on the st plane as a surface element with an *apparent* BRDF. Imagine the st 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. By recording the (apparent) BRDF of each st pixel (see Figure 2), we capture the aggregate reflectance of objects visible through that pixel window. The light vector L from the light source defines one direction of the BRDF while the (u, v, s, t) parameter implicitly defines the viewing direction V 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 data structure. 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.

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,

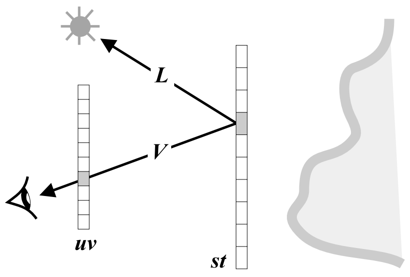


Fig. 2. Measuring the BRDF of the *pixel*.

```
For each viewpoint $(u, v)$
    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 $I_{u,v,\theta,\phi}$.
```

The parameter θ is the polar angle, and ϕ is the azimuth. The direction $(0, \phi)$ is parallel to the normal of the st plane. The parameters are localized to the light slab coordinate system, so transforming the light slab 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 viewpoint $(u, v)$
    For each pixel $(s, t)$
        $\rho(\theta, \phi)$ = pixel value at $(s, t)$ of $I_{u,v,\theta,\phi}$ / intensity of light source
```

One assumption is that there is no intervening medium, which absorbs, scatters or emits any radiant energy. Since the viewing direction of each pixel within one specific view is fixed, the BRDF ρ is simplified to a unidirectional reflectance distribution function which depends on the light vector only. Hence, the function ρ is parametrized by two parameters (θ, ϕ) only. There are two reasons we store the partial BRDF of each pixel in several fixed (u, v) views, instead of a complete BRDF for each pixel of a single st plane. Firstly, with this organization, the transformation and reconstruction of spherical harmonics (described in the next section) is simplified. Secondly, the reconstruction from spherical harmonic coefficients is performed only when the lighting changes. No reconstruction is needed when the user changes viewpoint. This is important for interactive display, since the user changes the viewpoint more often than the illumination.

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 pixel plane. This is because the actual object surface may not align with the pixel 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.

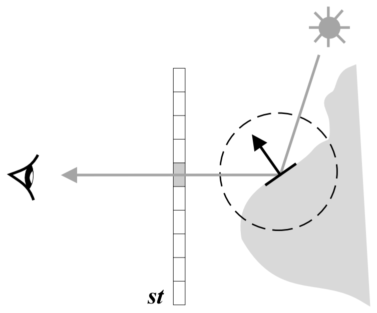


Fig. 3. The pixel plane may not be parallel with the object surface.

2.3 Spherical Harmonics

Storing the whole BRDF tables requires enormous storage space. For a single pixel, if the BRDF 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 256×256 image would require 100MB of storage.

To represent the BRDF more efficiently, the tabular data are transformed to spherical harmonic [7] domain. Spherical harmonics are analogous to the terms of a Fourier series, but 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.* [14] to model the entire range of incident angles. 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 the spherical harmonic domain using the following equations directly, without considering how to represent a bidirectional function as in [14].

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

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 $P_{0,0}(x) = 1$.

The $C_{l,m}$ are the coefficients of the spherical harmonics, which will stored for each pixel. The more coefficients used, the more accurate is the spherical harmonic representation. The 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 direction (θ, ϕ), solve

$$\rho(\theta, \phi) = \sum_{l=0}^{l_{max}} \sum_{m=-l}^{l} C_{l,m} Y_{l,m}(\theta, \phi). \tag{1}$$

for each pixel in each view, where $(l_{max})^2$ is the number of spherical harmonic coefficients to be used.

Figure 4 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 original distribution (left). The reconstructed distribution on the right is represented by only 25 spherical harmonic coefficients. Note that although we only sample the upper hemisphere of the reflectance distribution in this example, the reconstructed distribution on the right gives the whole spherical distribution. The distribution on the lower hemisphere is interpolated to prevent discontinuity (see section 3).

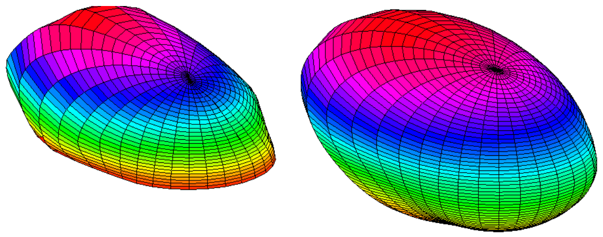


Fig. 4. Original sampled (left) and reconstructed (right) distribution.

2.4 Manipulating the Light Sources

Once the BRDFs are sampled and stored as spherical harmonic coefficients, they can be reconstructed and manipulated. The final radiance (or, simply value) of each pixel in each view is determined by evaluating Equations 1 and 2, given the intensity and the direction of the light sources.

$$\text{value at pixel } (s,t) \text{ in view } (u,v) = \sum_{i}^{n} \rho_{u,v,s,t}(\theta_i, \phi_i) I_i, \tag{2}$$

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.

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

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

where ρ_i^j is the reflectance of the j^{th} object illuminated by light L_i, and $\rho_i = \sum_{j=1}^{k} \rho_i^j$ is the aggregate reflectance recorded as the BRDF of the pixel. This superposition property is also pointed out by Nimeroff *et al.* [13] and Belhumeur and Kriegman [2].

Light Direction With Equation 2, the light direction can be changed by substituting a different value of (θ, ϕ). Figures 7(a) and (b) (see Appendix) show a teapot illuminated by a light source from the top and the right respectively.

Light Intensity Another manipulable parameter is the intensity I_i of the i^{th} light source. Figure 8(a) in the Appendix shows the Beethoven statue illuminated by a blue light from the left.

Multiple Light Sources We can arbitrarily add any number of light sources, at a cost in computation time. From Equation 2, a new reflectance ρ_i has to be evaluated using Equation 1 for each light source. Our current prototype can run at an acceptable interactive speed using up to 3 directional light sources. In Figure 8(b) (see Appendix), the Beethoven statue is illuminated simultaneously by a blue light from the left and a red light from the right.

Type of Light Sources Up to now, we have implicitly assumed that the light source for manipulation is directional. Directional light is very efficient in evaluating Equation 2, because all pixels in all views inside the same light slab will have the same (θ_i, ϕ_i) value. 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 2 using other types of light sources, since (θ_i, ϕ_i) will need to be recalculated from pixel to pixel.

Since the st plane where the pixels are located is not a fixed plane in 3D space (the coordinates (u, v, s, t) can only specify the direction), the intersected surface element that actually reflects the light may be located on any point on the ray $V(u, v, s, t)$ in Figure 5. 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 sources, 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 done easily for rendered images, real world scenes may be more difficult. Use of a range scanner may provide a solution. Figures 9(a) and (b) (see Appendix) show a cube 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 2.1, 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.

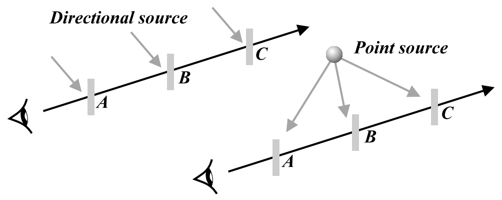


Fig. 5. Finding the correct light vector.

3 Discussion

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 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.

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. A natural scheme to compress the spherical harmonic coefficient is to drop the least significant coefficients. Moreover, we believe vector quantization is promising for further compression of the coefficient vectors.

Preventing Discontinuity While smooth functions can be represented by a small number of spherical harmonic coefficients, discontinuous functions require an infinite number of coefficients to represent, just like the Fourier series. Truncating the spherical harmonic series gives 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 6 and the right diagram in Figure 4). Another source of discontinuity is shadowing, which is unavoidable. Hard shadows will be smoothed out if represented by a finite sum of spherical harmonics (Figure 9 in Appendix).

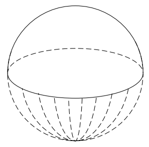


Fig. 6. The boundary value along the equator is linearly interpolated to prevent equatorial discontinuity in the sampled BRDF.

Independence of Sampled uv Resolution Whenever the viewer changes the lighting, a set of images viewed from some fixed 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.

4 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.

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

Another direction is to explore other representations than spherical harmonics, which are not efficient in representing the discontinuous functions. One potential scheme uses spherical wavelets. Belhumeur and Kriegman's model [2] is another efficient approach in representing convex diffuse objects. Its application to represent general objects requires further investigation. 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.

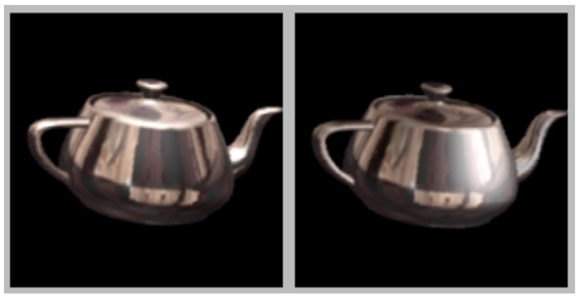
Acknowledgements

We are grateful to Prof. Tim Poston of Institute of Systems Science, National University of Singapore and Dr. Wai-Shing Luk of Katholieke Universiteit Leuven for their careful proofreading and helpful suggestions. We thank the reviewers for pointing out valuable references that we had overlooked. Thanks also to the light field project group of Stanford University for providing the `lifauth` package to the public. We modified the package to generate the data sets used in our prototype program. This work was supported by the CUHK Postgraduate Student Grants for Overseas Academic Activities.

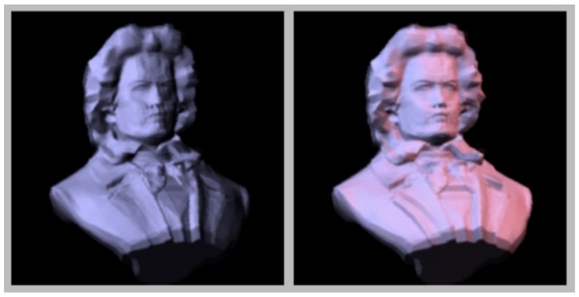
References

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.

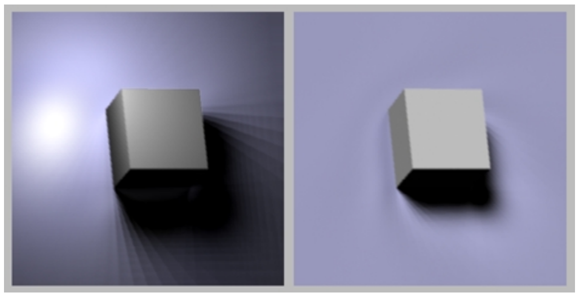
5. Shenchang Eric Chen and G. S. P. Miller. Cylindrical to planar image mapping using scanline coherence. United States Patent number 5,396,583, March 7 1995.
6. Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH '93 Proceedings)*, 1993.
7. 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. James T. Kajiya. Anisotropic reflection models. In *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 15–21, July 1985.
11. Marc Levoy and Pat Hanrahan. Light field rendering. In Computer Graphics *Proceedings, Annual Conference Series, SIGGRAPH'96*, pages 31–42, August 1996.
12. 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.
13. 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.
14. 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.
15. Gregory J. Ward. Measuring and modeling anisotropic reflection. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.



(a)Left: light from the above. (b)Right: light from the right. No. of spherical harmonics (s.h.): 25, st resolution (res.): 256x256. (Wong *et al.*, Fig. 7)



(a)Left: Beethoven statue illuminated by a single blue light from the left. (b)Right: red light added from the right. No. of s.h.: 25, st res.: 256x256. (Wong *et al.*, Fig. 8)



(a)Left: shadow cast by a point source. (b)Right: shadow cast by a directional source. No. of s.h.: 64, st res.: 256x256. (Wong *et al.*, Fig. 9)