Dynamic Shading in Image-Based Rendering

TR98-023
May 31, 1998

Manuel M. de Oliveira Neto

Gary Bishop

Graphics and Image Processing Laboratory
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3175

Dynamic Shading in Image-Based Rendering

Manuel M. de Oliveira Neto and Gary Bishop

Department of Computer Science
University of North Carolina at Chapel Hill
{oliveira, gb}@cs.unc.edu

Abstract

Image-based rendering techniques have been used to improve image quality, increase frame rate, and reduce modeling time. However, once the images have been acquired, they are bound to the light conditions of the scene during acquisition time. This paper describes how the Plenoptic Modeling approach can be extended to support applications that allow users to artificially change the original light conditions. A hybrid (geometric and image-based) solution is presented for producing shadows and view-dependent effects, such as specular highlights. The technique has been successfully tested with synthetic images and, under special lighting conditions, can also be applied to natural images.

1 Introduction

Image-based rendering is an interpolation technique [14] that uses pre-acquired images (either synthetic or natural), together with their corresponding camera parameters, and possibly additional related information, to generate new views of the environment. It tries to relax two principal difficulties of traditional three-dimensional computer graphics: modeling of complex environments, which is cumbersome and highly labor-intensive, and simulation of the intricate interaction between light and object surfaces. By using images as primitives, other benefits are also realized [7]: the display algorithms require modest computational resources, and the cost of interactively viewing the scene does not depend on its complexity. However, these benefits come at the cost of dealing with sampled representations of the original geometry [8], which, in turn, introduces new problems. Such problems are particularly noticeable when one tries to reconstruct new views of poorly sampled areas.

Image-based rendering techniques are usually based on the assumption that the light conditions of the scene are fixed. Except for image database approaches (Light Field [7], and Lumigraph [6]), view-dependent effects are not perceived as the observer moves. The Plenoptic Modeling [10] approach makes the extra assumption that scenes are composed of perfectly diffuse objects. This is unrealistic for many objects found in the real world, and puts some constraints on the optical properties of the objects that can appear in a scene.

This paper shows how changes in light conditions can be added to the Plenoptic Modeling paradigm by reshading pre-rendered images. A hybrid (geometric and image-based) solution is presented for producing shading, including shadows and first-order view-dependent effects, such as specular highlights. Such a strategy can be applied to synthetic images and, in the absence of shadows and important indirect lighting effects, to natural images too.

Section 2 describes relevant related work. Section 3 provides a brief overview of the Plenoptic Modeling [10] approach to image-based rendering. Section 4 describes our solution for producing shading effects in the context of Plenoptic Modeling. Section 5 shows results obtained with an implementation of the technique described. Finally, section 6 provides some conclusions.

2 Previous and Related Work

Wong et al. [19] showed how a Light Field or Lumigraph can be adapted to support changes in lighting conditions. In their approach, the apparent bidirectional reflectance distribution function (BRDF) of each pixel of the image database is computed using directional light sources[1]. Each view of the image database is rendered under different directional light and stored using the following algorithm [19]:

```
For each view point (u,v)
  For each directional light source's direction (theta, phi)
    Render the virtual scene
      For each pixel (s,t)
         brdf(theta, phi) = pixel value at (s,t)
```

In order to represent the BRDFs more efficiently (they use about 900 sampling light directions), the BRDFs are transformed to the spherical harmonic domain [5], and 16 to 25 coefficients of the spherical harmonic representation are stored per pixel [19]. For an image of size 256×256 pixels, where 25 floating point coefficients are used to represent each (R,G,B) channel, a total of 18.75 Mb of storage is required. The authors mention that, by storing a bitmap to indicate the nonzero coefficients and by storing only such coefficients, the storage requirement drops to 2 or 3 Mb per image [19].

Given the directions and intensities of the desired light sources, the stored coefficients are used to reconstruct the images of the database as they are needed. Such images remain valid until some light source direction or intensity is changed. The final views are displayed using the texture mapping technique of the Lumigraph system [6]. The approach can be extended to support other types of light sources, but it slows down the reconstruction of the database images. The need to control the lighting conditions under which the images are acquired constrains the use of this technique to synthetic environments. The use of a finite number of spherical harmonic coefficients smooths out sharp edges and shadows, since only smooth functions can be represented using a finite number of coefficients.

Fournier et al. [4] presented a framework for computing common illumination between real and synthetic scenes. In this approach, computer generated objects can be inserted into real scenes, casting shadows on (and being shadowed by) real objects. The authors make the distinction between "local" and "global" shadows, local meaning that objects are shadowing themselves or other close ones, and global meaning the shadows produced by global illumination. The approach can be divided into five major steps: approximation of scene objects wih boxes (based on measurements made on objects of the real scene), extraction of camera parameters of the original images, modeling of the real and synthetic light sources, insertion of a polygonal representation of the synthetic objects in the simplified scene, and use of classic radiosity computation to produce the final rendering. Rendering is done off-line. Because of the coarse modeling used, local shadows are ignored.

[1] The incident light direction is the same at any 3D point.

Drettakis et al. [3] improved the original work by Fournier et al. [4] using computer vision techniques to extract camera parameters and recover a more accurate 3D scene representation, and using radiosity-modulated textures to achieve interactive display rates. As in [4], this system also focused on global shadows of indoors environments.

The approach presented in this paper differs from the ones described in [4] and [3] by focusing on local instead of global shadows. Therefore, it is not suitable for computer augmented reality (CAR) applications. However, sharp shadows alone can still greatly improve the degree of realism of many applications. The simplicity of our approach makes it easy to implement, and suitable for interactive applications based on image warping.

3 Plenoptic Modeling

The Plenoptic Modeling [10] approach to image-based rendering is based on the notion of a Plenoptic Function [2] [1], and it can be applied to real and synthetic images. In order to simplify the discussion, this paper considers Plenoptic Modeling in terms of planar samples of the Plenoptic Function [11].

In a planar sample of the plenoptic function, each pixel of the image has associated a scalar value called *generalized disparity* δ (disparity, for short). Disparity values are obtained by computing $\delta = d/r$, where d is the distance from the center of projection to the (u, v) coordinates of the pixel in the image plane, and r is the distance from the center of projection to the corresponding sampled point in 3D space. Intuitively, disparity is a measure of how much the coordinates of a projected point change as the viewpoint moves. Planar samples of the plenoptic function can be used for reconstruction and resampling of new views of the scene.

The use of images extended with disparity assumes the ability to acquire depth on a per pixel basis. While depth acquisition from synthetic environments is a straightforward task, the same is not true for images acquired from the real world. Depth acquisition, however, is beyond the scope of this paper. The reader interested in the subject can find some useful references in [10].

New images can be obtained from planar samples by means of warping operations. The set of original planar samples is referred to as *reference images*, as opposed to *desired images*, which are the ones we want to synthesize. A forward warping maps pixels from a reference image to coordinates on the desired image plane, while an inverse warping maps pixel coordinates from a desired image to coordinates on the reference image plane [11]. Notice that although disparity values associated with pixels of a reference image can potentially be used to map pixels back to points in 3D space and reproject them into the desired image plane, the warping operations do not require these intermediate steps.

4 Light Effects for the Plenoptic Modeling Approach

This section describes how shadows and view-dependent effects can be incorporated into the Plenoptic Modeling approach. A technique is presented to

---

[2] A function that describes all information visible from each point in space, at a particular instant of time. It can be represented by $P(V_x, V_y, V_z, \theta, \phi)$, where the first three parameters correspond to the view position, while θ and ϕ define the view direction.

produce desired images with different shading than those of the reference ones. It is based on the reconstruction of 3D meshes that are used to compute shadows and estimate pixel normals. For each reference image, a mesh composed of connected components, each corresponding to a piece of continuous surface from the sampled geometry, is generated. Such components are identified using the rate of disparity change from pixel to pixel.

Mesh reconstruction and normal estimation are done as preprocessing. When a light source is moved, the corresponding shadow information is updated, remaining valid until the light source is moved again. In this approach, the reference images are extended with a normal vector per pixel, and with some connectivity information.

Section 4.1 presents an image warping framework for local illumination. It is intended to help the reader to understand some of the visibility and reconstruction problems associated with changes in shading of pre-rendered images, in particular, the difficulties associated with shadow computation.

4.1 An Image Warping Framework for Local Illumination

Figure 1 shows how a local illumination model can be applied using image warping. Each pixel of the reference image is simultaneously warped to the desired view, and to the six faces of the cube centered at a point light source. Notice that each face of the cube can be understood as a desired view (shadow buffer [17]). The viewer and light vectors used to compute shading are the vectors from the centers of projection to the coordinates of the pixels in the corresponding image planes. Thus, if for each pixel (u, v) corresponding to the projection of a point P in 3D space its normal (and, ideally, optical properties) is available, it is possible to generate new shaded views of the original image.

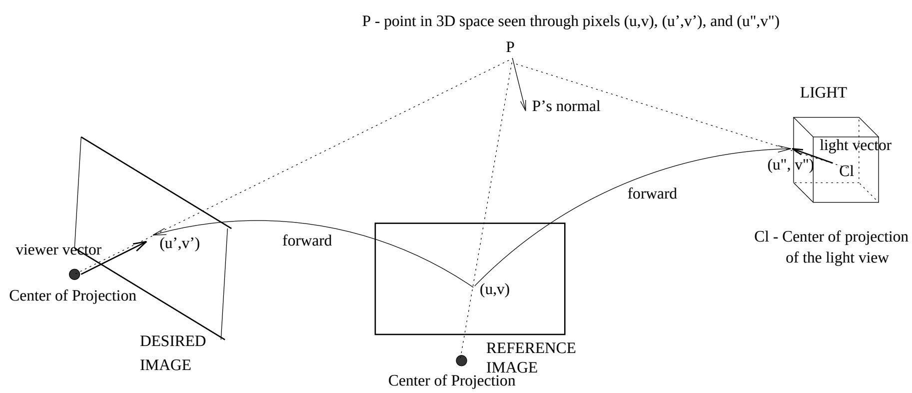

Fig. 1. Using image warping to produce desired views with different shading than that of the reference one, after adding a light source — a conceptual view.

Although conceptually correct, this framework has one drawback caused by the fact that pixels correspond to discrete samples of geometry. The spacing between the sampled points may produce leaks in the shadow buffers, *i.e.*, pixels that should be occluded appear as visible to the light source. Figure 2 depicts this situation. This problem can be solved by reconstructing a mesh that respects the connectivity of the surfaces represented by the pixels. A precise identification of which pixels belong to the same surface is critical for generating a mesh that avoids connecting pixels that should have no connection, as well as leaving pixels disconnected when they should be connected. A method for identifying the connectivity among pixels is needed. This is described next.

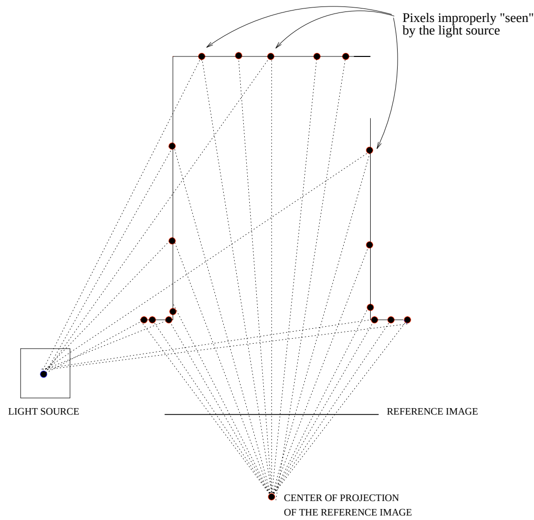

Fig. 2. The light source can see pixels that should be occluded by the actual surfaces. In this top view of a scanline, the solid lines represent geometry, while the dots correspond to the actual sampled 3D points.

4.2 Recovering Geometric Information from Range Images

The rate of change in disparity from pixel to pixel can be used to identify pixels that belong to the same surface. A segmentation process followed by the reconstruction of a geometric mesh can be used to produce shadow buffers without leaks, and to compute good approximations for pixel normals to be used for shading.

The disparity values of pixels associated with the projection of a line (or plane) onto the image plane vary linearly from pixel to pixel [13]. Thus, given three consecutive pixels corresponding to the projection of a line (or plane) onto the image plane, we have:

$$\delta_i - \delta_{i-1} = \delta_{i+1} - \delta_i \tag{1}$$

For curved objects, pixels can be seen as vertices of a piecewise bilinear approximation of their surfaces. For three consecutive pixels

$$\mid \delta_i - \delta_{i-1} \mid = \mid \delta_{i+1} - \delta_i \mid + \epsilon \tag{2}$$

where ϵ is a disparity threshold, can be used to search for discontinuities. If the disparities associated with two neighbor pixels do not satisfy equation 2, they are considered to belong to different surfaces [3].

As an object gets far away from the image plane, the corresponding disparity values approach zero. On the other hand, as the object gets closer to the image plane, the corresponding disparity values approach 1. For this reason, the disparity threshold (ϵ) must change adaptively.

This technique produces good segmentation of parts of images corresponding to different (pieces of the) objects. The images are divided into connected components, each of which corresponds to a piece of continuous surface from the sampled geometry. 3D meshes are generated by projecting the pixels back to 3D space, while keeping the separation among connected components. Such meshes can be further simplified to reduce the number of polygons associated with areas of the image where the curvature is almost constant, in a way similar to that used in [2]. Notice, however, that the mesh used in [2] consists of a single component.

A similar approach for object segmentation is used in [15], although for quite different purposes. In [15], multibaseline stereo [12] is used to produce dense depth maps, that are further converted into a triangular mesh. In order to detect discontinuities between surfaces corresponding to neighbor pixels, the authors use the following heuristic: look for triangles with large[4] difference in depth across any side [15]. A decimation process is also used to reduce the polygon count of the mesh. The resulting model is then texture-mapped and displayed. Notice that, in our approach, the possibly simplified meshes are only used for shadow computation. The geometry of the objects are preserved, since no changes are applied to the original disparity data.

Also, in our approach, normal vectors for each pixel can be approximated by averaging (before mesh simplification) the normals of triangles that share the vertex corresponding to the pixel. Notice that only triangles which belong to the same connected component as the pixel can contribute to its normal. Figure 3 shows images rendered using the technique described. Figure 3 (a) was used as input for mesh reconstruction and normal estimation. Figures 3 (b) and 3 (c) were rendered using the derived information. Notice that no light is subtracted from the reference images, since this would require knowledge about shape and position of the light sources. Although reference images rendered using only ambient light are more suitable for reshading, this is not a requirement. With the exception of Figure 3 (a), all reference images shown in this paper contain both ambient and diffuse components.

4.3 Shadows

Given the 3D meshes of all reference images, the geometry graphics pipeline can be used to render the view from the light source for each of the six faces of the cube into the OpenGL back buffer [18]. By precomputing the 3D coordinates corresponding to each pixel from the reference images, it is possible to map them to the light source space and mark whether they are visible or not. Notice that for static scenes this relationship remains valid until some light source position is changed. As the position of a light source changes, only the visibility information from that light source needs to be updated. Image warping can then be applied with the overhead of applying the illumination

[3] Configurations in which eq. 2 holds but pixels p_{i-1}, p_i, and p_{i+1} do not belong to the same surface are very rare and are usually associated with poorly sampled areas. Color comparison between neighbor pixels can be used to minimize errors in such cases.

[4] Unfortunately, no precise definition for large is provided.

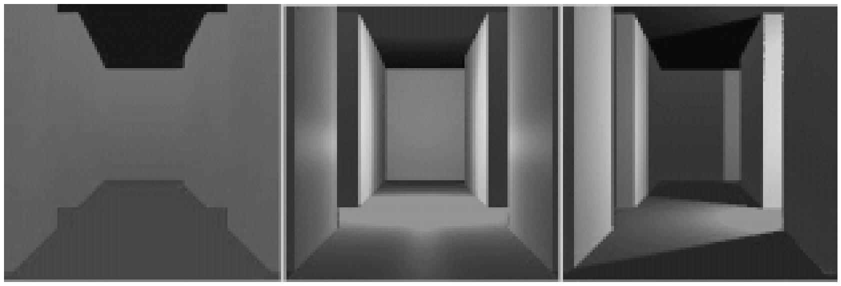

Fig. 3. (a) Reference image: maze rendered using only ambient light (brightened). (b) Application of a Phong illumination model to the image shown in (a) using the estimated normals. (c) Shadows.

Fig. 4. (a) Reference image. (b) and (c) Projected Shadows computed using the multiple connected component approach.

model per pixel. Figures 3 and 4 show examples of shaded images, including shadows, obtained with the technique described. For them, a single reference image was used.

Although the multiple connected component mesh technique used here is not guaranteed to produce correct shadows for an arbitrary configuration of viewing and light positions given a single reference image, it produces correct shadows if enough, appropriately segmented, reference images are available. Figures 7 to 10 illustrate this point. In general, however, single component meshes cannot produce correct shadows even if multiple reference images are used, because they improperly connect surfaces from different objects.

4.4 Shading Algorithm

The algorithm to reshade images with depth can be summarized as follows:

```
Preprocessing

(a) For each reference image
      Generate its multiple connected component mesh
      Compute pixel's 3D position
      Compute pixel's normals
      Simplify the mesh - optional

(b) For each light source
(b1)   Draw the 3D meshes to the back buffer
```

```
        Mark the pixels that are visible by the light source
Steps

    If light position has changed execute (b1)
    Warp the reference(s) image(s) to the desired view
    For each pixel of the desired image
      For each light source
         if pixel is lit, accumulate the current light source contribution
```

Algorithm 1. Reshading pre-rendered images

4.5 Reducing Shading Cost

To avoid applying the illumination model to each pixel of the final image for every frame, two alternative solutions can be used, although they have not been implemented:

1. One possibility is to render the current desired view, applying the illumination model on a per pixel basis. Such an image can be used to generate new views close to it. If the user moves more than a threshold from that particular view point (by translating or rotating), a new image is generated and used the same way. This solution, however, is dependent on the quality of the reconstructed image. Also, this approach introduces an overhead for computing new disparity values, which can be done with the extra cost of two multiplications, and two reciprocal operations per pixel [9].
2. Another possibility is to render desired images at the same position as the reference ones (in this case, no disparity computation is necessary) every time a light source is moved. Such images would contain only shadows, and ambient and diffuse components, and could be warped as a regular reference image.

5 Results

The algorithm described in section 4.4 was implemented and tested with synthetic images, since the generalized disparity values are readily available (for instance, they can be computed from the OpenGL depth buffer [18]). The implementation required only the inclusion of the preprocessing step and the shading function into a regular image warper code. Our current prototype was built on top of a non-optimized warper, and was used to generate the accompanying video.

Figures 3 and 4 show examples of images to which the algorithm has been applied. In these examples, a single reference image and an $\epsilon = 0.05$ was used to generate the 3D shadow mesh. Even small details are captured by the shadows (see the projection of the thumb on Figure 4 (c)). Figures 5 to 9 illustrate the use of multiple reference images to generate arbitrary shadows. Figure 5 shows two different views of the same teapot with a background plane. By adding a light source (figure 6), the shadow projected onto the plane is satisfactory for the case where the reference image and the view from the light source are similar (figure 6 (a)), while it is poor when they differ a lot (notice the projection of top of the lid in figure 6 (b)). By combining information from both images, the result shown in figure 7 is obtained. Using additional images to provide information about the spout and the handle (figure 8), the result presented in figure 9 is obtained. Although four reference images have been used to produce the desired shadow, the warping time is not affected, since the visibility from the light point of view has been established during preprocessing. If the light position is changed, however, new visibility relations need to be established.

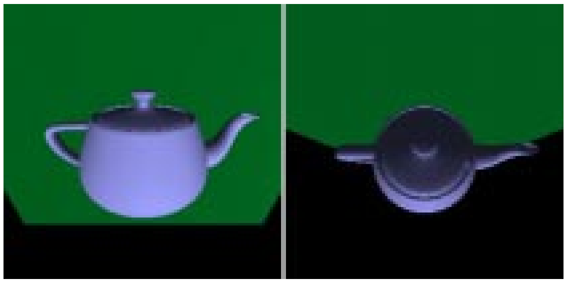

Fig. 5. Two views of the same teapot.

Fig. 6. Shadows generated using only information from the own image ($\epsilon = 0.05$).

Fig. 7. Shadow generated by combining information from the two images shown in figure 5 ($\epsilon = 0.05$).

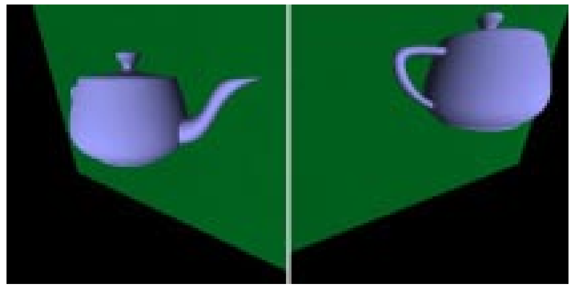

Fig. 8. Additional reference images providing information about the spout (a) and the handle (b).

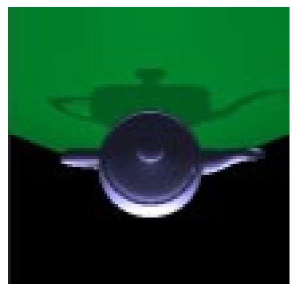

Fig. 9. Final image - shadow generated using information from four reference images ($\epsilon = 0.05$).

6 Conclusions

We have discussed how to extend the Plenoptic Modeling approach to support dynamic shading. A technique has been presented for computing shadows, based on the segmentation of the reference images into multiple connected components. Given an appropriately chosen set of reference images, it is possible to compute shadows that appear correct from any combination of view and light position. For still light sources, shadow computation can be regarded as preprocessing.

Some effort is currently being devoted to acquire depth from natural images. Some work using computer vision techniques is in progress, and a laser range finder has been acquired for this purpose. Under special lighting conditions, where shadows and important indirect lighting effects are avoided, our technique can be applied to natural images.

Acknowledgements

This work was supported by a fellowship from CNPq/Brazil under Process # 200054/95. Additional support provided by DARPA under order # E278 and NSF under grant # MIP-9612643.

References

1. Adelson, E., and Bergen, J. *The Plenoptic Function and the Elements of Early Vision.* Computational Models of Visual Processing. Chapter 1, Edited by Michael Landy and Anthony Moushon. The MIT Press, Cambridge, Mass. 1991.
2. Darsa, L., et al. *Navigating Static Environments Using Image-Space Simplifications and Morphing.* Proc. I3D (Providence, RI, April 27-30, 1997). In Interactive 3D Graphics Proceedings. pp. 25-34.
3. Drettakis, G., Robert, L., and Bougnoux, S. *Interactive Common Illumination for Computer Augmented Reality* . Proceedings of the 8th Eurographics Workshop on Rendering, St. Ettiene, France, June 1997.
4. Fournier, A., Gunawan, A., and Romanzin, C. *Common Illumination between Real and Computer Generated Scenes* . Proceedings of the Graphics Interface 93, pp. 254-262, Toronto, Ontario, Canada, 1993.
5. Glassner, A. *Principles of Digital Image Synthesis*, Morgan Kaufmann, San Francisco, CA, 1995, pp. 675-677.
6. Gortler, S., et al. *The Lumigraph.* Proc. SIGGRAPH 96 (New Orleans, LA, August 4-9, 1996). In Computer Graphics Proceedings. Annual Conference Series, 1996, ACM SIGGRAPH, pp. 43-54.
7. Levoy, M., Hanrahan, P. *Light Field Rendering.* Proc. SIGGRAPH 96 (New Orleans, LA, August 4-9, 1996). In Computer Graphics Proceedings. Annual Conference Series, 1996, ACM SIGGRAPH, pp. 31-42.
8. Levoy, M. *Expanding the Horizons of Image-Based Modeling and Rendering*, Panel on Image-Based Rendering - slide set. SIGGRAPH 97. URL http://graphics.stanford.edu/talks/imbased/slides.
9. Mark, W. *3D Warping in* R^3. Internal Publication. Department of Computer Science, University of North Carolina at Chapel Hill, September 1997.
10. McMillan, L., Bishop, G. *Plenoptic Modeling: An Image-Based Rendering System.* Proc. SIGGRAPH 95 (Los Angeles, CA, August 6-11, 1995). In Computer Graphics Proceedings. Annual Conference Series, 1995, ACM SIGGRAPH, pp. 39-46.
11. McMillan, L. *An Image-Based Rendering Approach to Three-Dimensional Computer Graphics.* Ph.D. Dissertation, UNC-CH, 1997.
12. Okutomi, M, and Kanade, T. *A Multiple-Baseline Stereo.* IEEE Transactions on Patern Analysis and Machine Intelligence, 15(4), 1993, pp. 353-363. Science, University of North Carolina at Chapel Hill, July 1997.
13. Popescu, V. Unpublished manuscript. Department of Computer Science, University of North Carolina at Chapel Hill, July 1997.
14. Pulli, K., et al. *View-Based Rendering: Visualizing Real Objects from Scanned Range and Color Data.* Proceedings of the 8th Eurographics Workshop on Rendering, St. Ettiene, France, June 1997.
15. Rander, P., Narayanan, P., and Kanade, T. *Virtualized Reality: Constructing Time-Varying Virtual Worlds From Real World Events.* Proceedings of the IEEE Visualization'97, Phoenix, Arizona, October 19-24, 1997, pp. 277-283.
16. Sillion, F. et al. *A Global Illumination Solution for General Reflectance Distributions.* Proc. SIGGRAPH 91 In Computer Graphics Proceedings. Annual Conference Series, 1991, ACM SIGGRAPH, pp. 187-196.
17. Williams, L. *Casting Curved Shadows on Curved Surfaces.* Proc. SIGGRAPH 78. In Computer Graphics Proceedings. Annual Conference Series, 1978, ACM SIGGRAPH, pp. 270-274.
18. Woo, M., Neider, J., and Davis, T. *OpenGL Programming Guide.* Second Edition. Addison-Wesley, 1997.

19. Wong, T. et al. *Image-based Rendering with Controllable Illumination.* Proceedings of the 8th Eurographics Workshop on Rendering, St. Ettiene, France, June 1997. pp. 13-22.