Appearance Modeling for Mixed Reality: Photometric Aspects

Katsushi Ikeuchi Yoichi Sato Ko Nishino Imari Sato

Institute of Industrial Science
The University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo, Japan 106-8558
ki@iis.u-tokyo.ac.jp

Abstract

This paper presents an overview of our efforts on mixed reality. These efforts span two aspects: how to create models of virtual objects, and how to integrate such virtual objects with real scenes. For model creation, we have developed two methods, model-based and eigen-texture rendering methods, both of which automatically create models by observing the real objects. For the integration of a virtual object with a real scene, we have developed a method that renders virtual objects based on real illumination distribution. We have successfully tested the proposed methods by using real images to demonstrate their effectiveness.

1 Introduction

Virtual reality is one of the most important techniques for human-computer communications. In particular, mixed reality, which integrates virtual images of objects with real background-images, is the key technique for electronic shopping and the virtual museum. Main research issues in mixed reality include how to obtain virtual images of objects, and then how to integrate them with real images.

Our efforts in the mixed reality research span two aspects: how to create models of virtual objects and how to integrate such virtual objects with real scenes. For model creation, we have developed two methods, model-based and eigen-texture rendering, both of which automatically create such rendering models by observing real objects. The model-based rendering method first analyzes input images of real objects, obtains reflectance parameters from this analysis, and then, using the determined reflectance parameters, generates the virtual image [18]. This method stores very compact information, namely reflectance parameters, and works well when an object surface follows a known reflectance model.

For other objects that do not follow simple reflectance models, we have developed the eigen-texture-rendering method [12]. This method samples appearances of a real object, pastes the images onto the surface of the 3D model of the object, and then compresses them on the 3D surface (not on the image plane). Later, the compressed data can be used for generating a virtual image of the object. The method does not assume any known reflectance model, nor does it require any detailed reflectance analysis, as was the case in model-based rendering. Using the 3D model of the object, the method can also generate shadows that appear to have been cast by the virtual object, something that the image-based rendering fails to do. Moreover, the method achieves a high compression ratio because the compression is performed along pixels corresponding to the same physical points.

For integration of virtual object with real scenes, we have developed a method that renders virtual objects based on real illumination distribution [15]. First, a radiance distribution in the real scene is determined from two omni-directional images of the scene. Then the measured radiance distribution is used for rendering virtual objects superimposed onto the scene image. The proposed method has the ability to synthesize a convincing image even for a complex radiance distribution, a task at which other methods often fail.

2 Creating models from observation

Currently, most VR (virtual reality) systems utilize image-based rendering [2,5,10]. The image-based rendering samples a set of color images of a real object and stores them on the disk of a computer. A new image is then synthesized either by selecting an appropriate image from the stored set or by interpolating multiple images. Apple's QuickTime VR is one of the earlier successful image-based rendering methods. Image-based rendering does not assume any reflectance characteristics of objects nor does it require any detailed analysis of the reflectance characteristics of the objects; rather, the method need only to take images of an object. The method can be applied to a wide variety of real objects. And because it is also quite simple and handy, image-based rendering is ideal for displaying an object as a stand-alone without any background for the

0-7803-5731-0/99/$10.00 ©1999 IEEE

virtual reality.

On the other hand, image-based methods have critical disadvantages on application to mixed reality. Few image-based rendering methods employ accurate 3D models of real objects. Thus, it is difficult to make cast shadows under real illuminations corresponding to the real background-image.

Unlike image-based rendering, model-based rendering assumes reflectance models of an object and determines reflectance parameters through detailed reflectance analysis. Later, the method uses those reflectance parameters to generate virtual images by considering illumination conditions of the real scene. Since the reflectance parameters are obtained at every surface point of the object, integration of synthesized images with the real background can be accomplished quite realistically, i.e., the method can generate a realistic appearance of an object as well as of the shadows cast by that object onto the background.

2.1 Model-based rendering

We have developed a model-based rendering method for modeling object reflectance properties, as well as object shapes, by observing real objects [4]. First, an object surface shape is reconstructed by merging multiple range images of the object. By using the reconstructed object shape and a sequence of color images of the object, parameters of a reflection model are estimated in a robust manner. The key point of the proposed method is that, first, the diffuse and specular reflection components are separated from the color image sequence, and then, reflectance parameters of each reflection component are estimated separately. This approach enables estimation of reflectance properties of real objects whose surfaces show specularity as well as diffusely reflected lights. The recovered object shape and reflectance parameters are then used for synthesizing object images with realistic shading effects under arbitrary illumination conditions.

Fig. 1 shows two frames of the input color image sequence as well as two synthesized images that were generated using the same illuminating/viewing condition as the input color images. It can be seen that the synthesized images closely resemble the corresponding real images. In particular, highlights, which generally are a very important cue of surface material, appear on the side and the handle of the mug naturally in the synthesized images.

2.2 Eigen-texture rendering method

The model-based rendering method can generate realistic appearances of an object from reflectance parameters determined from real objects. Unfortunately, however, some classes of object surfaces reveal complicated reflectance models, and the model-based rendering method cannot be applied to those classes of objects.

For those object surface classes, we propose a new rendering method, which we refer to as the eigen-texture rendering method [12,13,22]. Fig. 2 displays an overview of the proposed method. The eigen-texture rendering method creates a 3D model of an object from a sequence of range images. The method aligns and pastes color images of the object onto the 3D surface of the object model. Then, the method compresses those images in the coordinate system defined on the 3D-model surface. This compression is accomplished using the eigenspace method. The synthesis process is achieved using the inverse transformation of the eigenspace method. Cast shadows are generated using the 3D model. A virtual image under a complicated illumination condition is generated by summation of component virtual images sampled under single illuminations thanks to the linearity of image brightness.

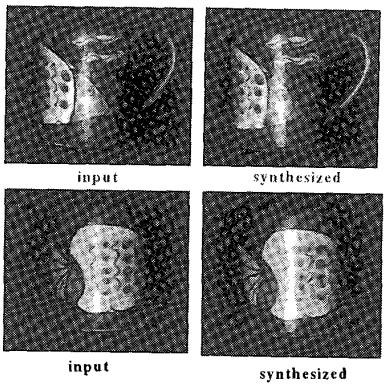


Fig. 1 Input and synthesized images

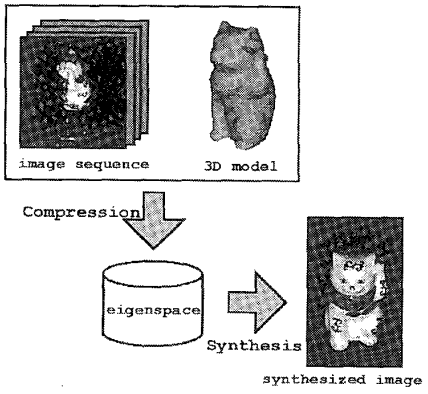


Fig. 2 Eigen-texture rendering method

We have implemented the eigen-texture rendering method, and have applied it to a real object. The same experimental setup in the model-based rendering experiment is used to sample sequence of color and range images of a real object. Fig. 3 shows the images synthesized by using 8 dimensional eigenspace. Details of the images are reconstructed accurately, and the synthesized images are indistinguishable from the input images. The compression ratio of the input image sequence was 7.9%. This ratio is quite

high when compared with that of other image compression methods. Experimental result proves that the eigen-texture rendering method is sufficiently effective in terms of compression. This is because the eigen-texture rendering method compresses the sequence of cell images that come from the same physical area on the object surface. The variances in the cell images are only caused by the relative directional change of the light source and the position of the viewer. Cell images are thus highly correlative. On the other hand, current image-based rendering methods usually compress images in the image coordinate systems. Each pixel in the sequence corresponds to a different physical point in the image sequence. Thus, the variance carries both those from the imaging geometries and those from the physical locations.

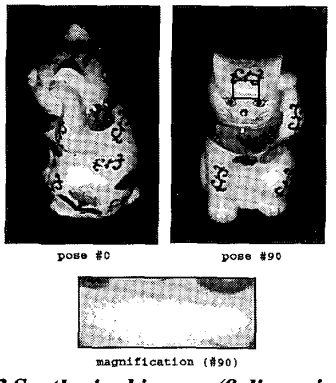


Fig 3 Synthesized images (8 dimensions)

Using the model-based and eigen-texture rendering method, we can create appearances of a virtual object. The next section considers how to integrate these appearances with real scenes for mixed reality.

3 Integrating virtual objects with a real scene

For superimposing virtual objects onto a real scene appropriately [1,20], the following three aspects have to be taken into account: geometry, illumination, and time. More specifically, the virtual object has to be located at a desired location in the real scene, and the object must appear at the correct location in the image (*consistency of geometry*). Also, shading of the virtual object has to match that of other objects in the scene, and the virtual object must cast a correct shadow, i.e., a shadow whose characteristics are consistent with those of the shadows in the real scene (*consistency of illumination*). Lastly, motions of the virtual object and the real objects have to be coordinated (*consistency of time*).

In this paper, we are mainly concerned with illumination consistency. Illumination of the real scene is directly measured and used for rendering virtual objects [15]. Our method measures a radiance distribution from all directions by a pair of omni-directional images taken by a CCD camera with a fisheye lens. If we use only one omni-directional image, we can determine only the radiance distribution seen from the particular point where the omni-directional image was captured. To overcome this limitation, our method uses an omni-directional stereo algorithm for measuring a radiance distribution of the real scene as a 3D spatial distribution (3D triangular mesh). Once the radiance distribution is obtained as a triangular mesh, the real radiance distribution can be used for rendering a virtual object and for generating shadows cast by the virtual object onto the real scene wherever the virtual object is placed in the scene.

3.1 Constructing a radiance map using omni-directional Stereo

A CCD camera with a fisheye lens is placed at two known locations in the scene to capture two omni-directional images from different locations. The direction of line-of-sight is determined from the image coordinate of the point corresponding to the line-of-sight using a known geometry of the fisheye lens.

Most of the incoming light energy in a real scene comes from direct light sources such as a fluorescent lamp and a window to the outside; the rest of the incoming light energy comes from indirect illumination such as reflection from a wall. For this reason, it is important to know the accurate locations of direct light sources in order to represent an illumination distribution of a real scene.

Fortunately, direct light sources usually appear as significantly bright points in an omni-directional image. Therefore, it is relatively easy to identify direct light sources in the image. In our method, we extract feature points with high contrast in the two omni-directional images by using a corner extractor.

After feature points are extracted, 3D coordinates of points in the real scene corresponding to the extracted feature points are determined by using a stereo algorithm. This is done by obtaining an intersection point of a pair of lines-of-sight given by a pair of feature points.

In order to include indirect light sources such as walls and ceilings, we set up a 3D triangular mesh of the entire scene. First, we construct a 2D triangular mesh by applying 2D Delaunay triangulation to the feature points in the first omni image. That determines the connectivity of a 3D triangular mesh whose vertices are the 3D points corresponding to the feature points in the image. Then, using the connectivity, a 3D triangular mesh is created from the 3D feature points.

The obtained triangular mesh approximates an entire

shape of the real scene, e.g., the ceiling and walls of a room, which act as direct or indirect light sources. After the shape of the real scene is obtained as a triangular mesh, the radiance of the scene is estimated by using the image brightness of the omni-directional images. This brightness value is warped on the corresponding triangular mesh. The 3D textured mesh is referred to as a radiance map.

3.2 Generating soft shadow

For superimposing virtual objects onto the real scene, we first compute the total irradiance at a point on a virtual object surface from the radiance map. Then, we also compute the partial irradiance due to the occlusion by a virtual object. By multiplying the ratio of these two irradiances to the real irradiance, we generate soft shadow. In this section, we assume that a virtual object always exists between the camera projection center and real objects in the scene and that the virtual object is located on the table, of which the position is known.

For each pixel in the input image of the real scene, a ray extending from the camera projection center through the pixel is generated by using the transformation between the world coordinate system and the image coordinate system. If the ray intersects a virtual object, we consider that the pixel corresponds to a point on the virtual object surface. Then we compute a color to be observed at the surface point under the measured radiance distribution of the real scene with the Torrance-Sparrow reflectance model. The computed color is stored in the pixel as the surface color of the virtual object at the pixel.

If a ray through an image pixel does not intersect with a virtual object, the color of a real object surface corresponding to the image pixel needs to be modified so that a shadow cast by the virtual object is created on the real object surface.

First, we obtain a 3D coordinate of a point on a real object surface where a ray through an image pixel intersects the real object. Then, a total irradiance E_1 at the surface point is computed from the radiance map by assuming that a virtual object does not occlude incoming light at the surface point. Then, a partial irradiance E_2 at the surface point given by non-occluded light sources is computed as

$$E_2 = \sum_{i=0}^{N} (2\pi/N) S_i L(\theta_i, \varphi_i) \cos\theta_i$$

where $S_i = 0$ if the virtual object occludes $L(\theta_i, \varphi_i)$, and $S_i = 1$ otherwise.

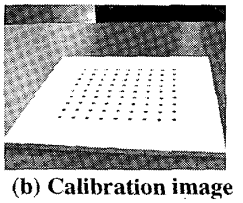
The ratio of the total radiance E_2 to the total radiance E_1 is multiplied to the color at the intersection between the ray and the plane on which the case shadow is generated.

3.3 Experimental Results

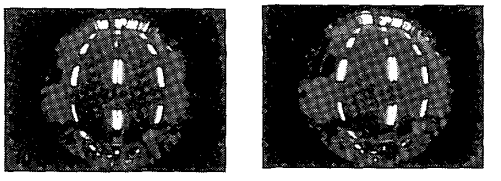
An image was taken of a tabletop and miscellaneous objects on the tabletop in our laboratory. From the same camera position, another image of the tabletop with a calibration board was taken. The input image and the calibration image are shown in Figure Fig. 4 (a) and (b).

First, regularly spaced dots on the calibration board were extracted in the calibration image to determine their 2D image coordinates. From pairs of the 2D image coordinates and the 3D world coordinates that were given a prior, the transformation between the world coordinate system and the image coordinate system was estimated by using the camera calibration algorithm [21]. Two omni-directional images of the scene, e.g., the ceiling of the laboratory in this experiment, were taken. Fig. 4 (c) shows the two omni-directional images. From this pair of images, the radiance map shown in Fig 4 (d) is obtained using the omni-stereo algorithm. Fig 4 (e) shows the resulting image that contains a virtual object with a soft shadow on the tabletop.

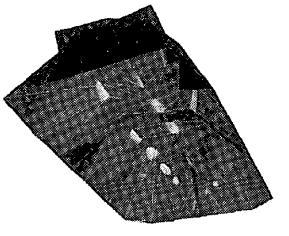
(a) Original scene



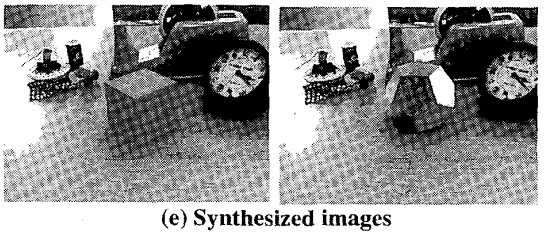
(b) Calibration image



(c) A pair of omni images



(d) Measured radiance map



(e) Synthesized images

Fig 4 Results of illumination estimation

4 Estimating illumination distribution from a single image

The previous method requires direct measurements of illumination distributions using a pair of fish-eye cameras. Some applications require to estimate illumination distributions from a single image. This section describes a method to estimate such distributions from observing a distribution of shadows around an object of which shape is known. Let us assume that the surface on which an object exists, be a Lambertian surface.

Let us examine the terms in the equations to calculate soft-shadows. Since the shape of an object is known, occlusion parameters S_i , whether the direction, *i*, is occluded or not, is given from the relation between the pixel position and the object shape. Once the sampling direction is determined, $\cos\theta_i$ is also a known value. These two known parameters are combined into a known parameter a_i . E_i is the image brightness at the pixel. Here, ^{L}i is an unknown illumination brightness. Thus,

$$E = \sum_{i=0}^{N} a_i L_i$$

is a linear equation of unknown value L_i , where

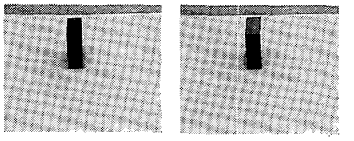
$$a_i = (2\pi/N) S_i \cos\theta_i ,$$
$$L_i = L(\theta_i, \varphi_i) .$$

We can set up one equation at each pixel. From observing a distribution of image brightness inside of a soft shadow E_j , we can set up simultaneous linear equations:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{M1} & a_{M2} & \cdots & \cdots & a_{MN} \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ \\ L_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \\ E_M \end{bmatrix}$$

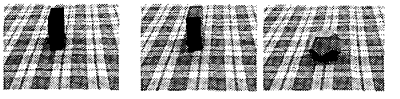
By solving this set of equations, we can estimate an illumination distribution from a single image. From observing image brightness distribution inside of a soft shadow in Fig 5a, we obtain an illumination distribution, and then synthesized a soft shadow in Fig. 5b.

In case of an unknown albedo, the method requires a pair of images, with and without an object.Figure 6a shows one of a pair of an input image. First, we compensate the effect of albedo by taking the ratio of image brightness, between these two images, at each pixel. Then, using the ratio distribution, we set up a similar set of simultaneous linear equations. By solving these equations, we can estimate illumination distribution. From the estimated distribution, we can obtain soft shadows. Figure 6b shows those obtained from Figure 6a.



(a) Original (b) Synthesized

Fig 5 Known Albedo



(a) Original (b) Synthesized

Fig 6 Unknow Albedo

5 Conclusions

We have explored the automatic generation of photorealistic object models from observation. For those objects with known reflectance models, we proposed a model-based rendering method that obtains reflectance parameters from sequences of range and color images. For other classes of objects, we have developed an eigen-textured rendering method that stores and compresses appearances of objects on their 3D mesh models. Our experiments have shown that both of our rendering methods can be effectively used for synthesizing realistic object images. We have also developed a new method of superimposing virtual objects onto images of real scenes by taking into account the radiance distribution of the scenes. To demonstrate the effectiveness of the proposed method, we have successfully tested our method by using real images taken in both indoor and outdoor environments.

As a future direction, we begin a project to obtain virtual models of historic assets in Japan. We, Japanese, is a unique position in this respect, because most of Japanese cultural assets are made of wood and papers, while most of foreign assets are made of stones. Thus, at any moments, they may be destroyed due to thunderstorm or fire. Figure 7 shows one of the digitized image of Kamakura big Buddha. We are currently working on this set of data into a

mixed reality model.

Fig 7 Range data of Kamakura Big Buddda

Acknowledgment

This work is partially supported by Monbusho Shin-Pro 09NP1401 and Monbusho Kakenhi 09450164. Thanks also go to CAD center and Kotokuzi-temple to digitize the big Buddha.

References

[1] R. Azuma, "A survey of augmented reality, " *Presence*, vol. 6, no. 4, pp. 355-385, August 1997.

[2] M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging virtual objects with the real world," *Proc. SIGGRAPH 92,* pp. 203-210.

[2] S. Chen, "Quicktime VR," *Proc. SIGGRAPH 95,* pp.29-38.

[3] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," *Proc. SIGGRAPH 96,* pp. 303-312.

[4] A. Fournier, A. Gunawan and C. Romanzin, "Common Illumination between Real and Computer Generated Scenes," *Proc. Graphics Interface '93*, pp.254-262.

[5] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," *Proc. SIGGRAPH 96,* pp. 43-54.

[6] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *Proc. SIGGRAPH 92*, pp. 71-78.

[7] K. Ikeuchi and K. Sato, "Determining reflectance properties of an object using range and brightness images," *IEEE Trans. PAMI,* 13(11): 1139-1153, 1991.

[8] G. Kay and T. Caelli, Inverting an illumination model from range and intensity maps, *CVGIP: IU,* vol. 59, pp. 183-201, 1994.

[9] G. K. Klinker, S. Shafer, and T. Kanade, The measurement of highlights in color images, *Int. J. Comp. Vision*, 2: 7-32, 1988.

[10] M. Levoy and P. Hanrahan, "Light field rendering," *Proc. SIGGRAPH 96*, pp. 31-42.

[11] S. Nayar, K. Ikeuchi, and T. Kanade, "Surface reflection," *IEEE Trans. PAMI*, 13(7): 611-634, 1991.

[12] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-texture method," IEEE Conf CVPR, pp.618-622, June 1999.

[13] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle, "View-based rendering," *Proc. 8th Eurographics Workshop on Rendering,* June 1997.

[14] K. Torrance and E. Sparrow, "Theory for off-specular reflection from roughened surface," *J. Opt. Society of America,* 57: 1105-1114, 1967.

[15] I. Sato, Y. Sato, and K. Ikeuchi, "Acquiring a radiance distribution to superimpose virtual objects onto a real scene," IEEE Trans Visualization and Computer Graphics, Vol. 5, No. 1, pp.1-12, January 1999.

[16] I. Sato, Y. Sato, and K. Ikeuchi, "Illumination distribution from shadows," IEEE Conf. CVPR, pp.306-312, June 1999.

[17] Y. Sato and K. Ikeuchi, "Temporal-color space analysis of reflection," *J. Opt. Society of America A*, 11(11): 2990-3002, November 1994.

[18] Y. Sato, M. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," *Proc. SIGGRAPH 97*, pp.379-387.

[19] M. Wheeler, Y. Sato, and K. Ikeuchi, "Consensus surfaces for modeling 3D objects from multiple range images," *Proc. 6th Int. Conf. Comp. Vision*, pp. 917-924, 1998.

[20] A. State, G. Hirota, D. Chen, W. Garrett, and M. Livingston, "Superior augmented-reality registration by integrating landmark tracking and magnetic tracking," *Proc. SIGGRAPH 96,* pp. 429-438, August, 1996.

[21] R. Tsai, "A Versatile Camera Calibration Technique for High Accuracy Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE J. R&A*, vol. 3, no. 4, pp. 323-344, 1987.

[22] Z. Zhang, "Modeling geometric structure and illumination variation of a scene from real images," *Proc. 6th Int. Conf. Comp. Vision,* pp.1041-1046, 1998.