Eigen-Texture Method : Appearance Compression based on 3D Model

Ko Nishino Yoichi Sato Katsushi Ikeuchi
Institute of Industrial Science, The University of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106-8558, Japan
{kon,ysato,ki}@cvl.iis.u-tokyo.ac.jp

Abstract

Image-based and model-based methods are two representative rendering methods for generating virtual images of objects from their real images. Extensive research on these two methods has been made in CV and CG communities. However, both methods still have several drawbacks when it comes to applying them to the mixed reality where we integrate such virtual images with real background images. To overcome these difficulties, we propose a new method, which we refer to as the Eigen-Texture method. The proposed method samples appearances of a real object under various illumination and viewing conditions, and compresses them in the 2D coordinate system defined on the 3D model surface. The 3D model is generated from a sequence of range images. The Eigen-Texture method is practical because it does not require any detailed reflectance analysis of the object surface, and has great advantages due to the accurate 3D geometric models. This paper describes the method, and reports on its implementation.

1 Introduction

Recently, there has been extensive development of mixed reality systems for the purpose of integrating virtual images of objects with real background images. Main research issues include how to obtain virtual images of objects and then seamlessly integrate those objects with real images. There are two representative rendering methods to obtain such virtual images from real objects: image-based and model-based.

The image-based rendering samples a set of color images of a real object and stores the images on the disk of a computer[3, 4]. A new image is then synthesized either by selecting an appropriate image from the stored set or by interpolating multiple images[1]. As 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, 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 virtual reality. On the other hand, image-based methods have disadvantages for application to mixed reality. Few image-based rendering methods employ acccurate 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[11, 12]. Later the method uses those reflectance parameters to generate virtual images by considering illumination conditions of the real scene. By using these reflectance parameters, integration of synthesized images with the real background can be accomplished quite realistically[9]. However, model-based rendering has nontrivial intrinsic constraints; it cannot be applied to objects whose reflectance properties cannot be approximated by using simple reflection models.

To overcome the problems posed by the previous methods, we propose a new rendering method, which we refer to as *Eigen-Texture method*. Figure 1 displays an overview of the proposed method. The *Eigen-Texture 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, it compresses those appearances in the 2D 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. Virtual images under a complicated illumination condition can be generated by summation of component virtual images sampled under single illuminations thanks to the linearity of image brightness.

Related work has been done by Zhang[17], who uses a stereo camera technique to obtain a partial 3D model of an object, and compresses the pixel values of each point of the object surface with respect to the image coordinates. Our method differs from his method in that we can generate a full 3D model of the object with all appearances from 360 degrees viewer directions; this ability gives us great advan-

1063-6919/99 $10.00 (c) 1999 IEEE

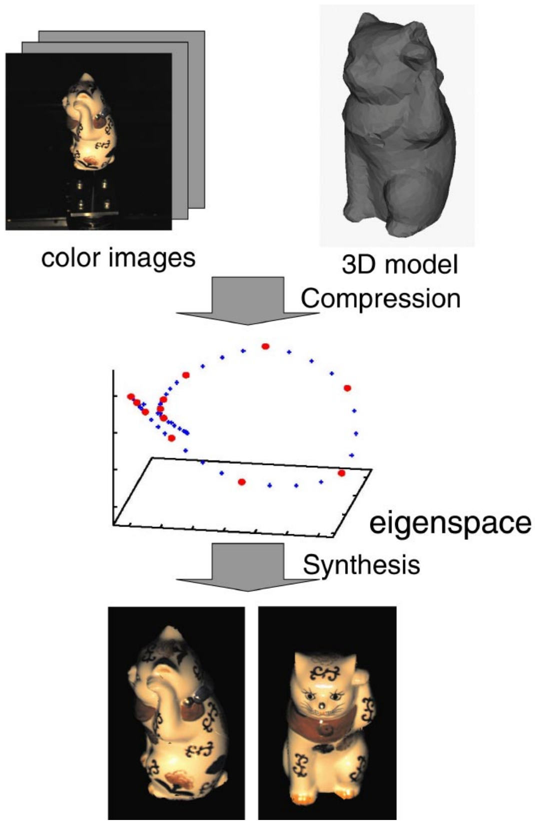

Figure 1: Outline of the *Eigen-Texture method*.

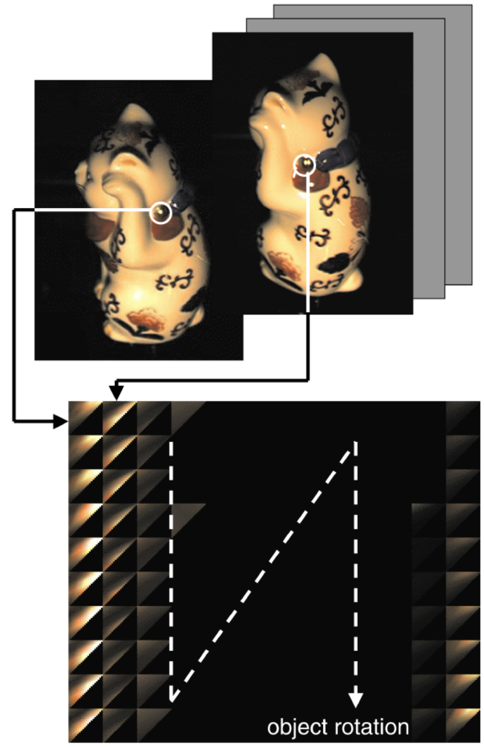

Figure 2: A sequence of cell images.

tages in mixed reality systems, e.g., making cast-shadows.

The remainder of the paper is organized as follows. In Section 2, we describe the *Eigen-Texture method*. In Section 3, we describe the implementation of the proposed method and discuss the results of the experiments we conducted to demonstrate the efficiency of the proposed method. In Section 4, we describe the results of applying our method in integrating virtual images into real scenes. In section 5, we discuss the merits and limitations of our method.

2 Eigen-Texture Method

This section describes the theory of the *Eigen-Texture method*. The method samples a sequence of color and range images. Once these two sequences are input to the system, a 3D geometric model of the object is created from the sequence of range images. Each color image is aligned with the 3D model of the object. In our system, this alignment is relatively simple because we use the same color CCD camera for taking both range and color images. For other systems, camera calibration may be required to determine the relative relation between two cameras. Each color image is divided into small areas that correspond to triangle patches on the 3D model. Each triangle patch is normalized to have the same shape and size as that of the others. Color images on the small areas are warped on a normalized triangular patch. This paper refers to this normalized triangular patch as a *cell* and to its color image as a *cell image*. A sequence of cell images from the same cell is collected as shown in Figure 2. Here this sequence depicts appearance variations on the same physical patch of the object under various viewing conditions. These cell images corresponding to each cell are compressed using the eigenspace method. Note that the compression is done in a sequence of cell images, whose appearance change are due only to the change of brightness. Thus, high compression ratio can be expected with the eigenspace method. Furthermore, it is possible to interpolate appearances in the eigenspace.

Eigenspace compression on cell images can be achieved by the following steps.

The color images are represented in RGB pixels with 24-bit depth, but the compression is accomplished in YC_rC_b using $4:1:1$ subsampling. First, each cell image is converted into a $1 \times 3N$ vector $\mathbf{X}_m$ by arranging color values for each color band YC_rC_b in a raster scan manner (Eq.1). Here, M is the total number of poses of the real object, N is the number of pixels in each cell image and m is

1063-6919/99 $10.00 (c) 1999 IEEE

the pose number.

$$\mathbf{X}_m = \left[\begin{array}{ccccc} x^Y_{m,1} & \ldots & x^{C_r}_{m,1} & \ldots & x^{C_b}_{m,N} \end{array} \right] \quad (1)$$

Then the sequence of cell images can be represented as a $M \times 3N$ matrix as shown in Eq.2.

$$\mathbf{X} = \left[\begin{array}{cccc} \mathbf{X}^T_1 & \mathbf{X}^T_2 & \ldots & \mathbf{X}^T_M \end{array} \right]^T \quad (2)$$

The average of all color values in the cell image set is subtracted from each element of matrix $\mathbf{X}$. This ensures that the eigenvector with the largest eigenvalue represents the dimension in eigenspace in which the variance of images is maximum in the correlation sense.

$$\mathbf{X}_a = \mathbf{X} - \left[\begin{array}{ccc} E & \ldots & E \\ . & \ldots & . \\ E & \ldots & E \end{array} \right] \quad (3)$$

$$E = \frac{1}{3MN} \sum_{i=1,j=1,c\in\{Y,C_r,C_b\}}^{M\ N} x^c_{i,j}$$

With this $M \times 3N$ matrix, we define a $3N \times 3N$ matrix $\mathbf{Q}$, and determine eigenvectors $\mathbf{e}_i$ and the corresponding eigenvalues λ_i of $\mathbf{Q}$ by solving the eigenstructure decomposition problem.

$$\mathbf{Q} = \mathbf{X}^T_a \mathbf{X}_a \quad (4)$$

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \quad (5)$$

At this point, the eigenspace of $\mathbf{Q}$ is a high dimensional space, i.e., $3N$ dimensions. Although $3N$ dimensions are necessary to represent each cell image in a exact manner, a small subset of them is sufficient to describe the principal characteristics and enough to reconstruct each cell image with adequate accuracy. Accordingly, we extract k ($k \ll 3N$) eigenvectors which represent the original eigenspace adequately; by this process, we can substantially compress the image set. The k eigenvectors can be chosen by sorting the eigenvectors by the size of the corresponding eigenvalues, and then computing the eigenratio (Eq.6).

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{3N} \lambda_i} \geq T \quad where\ \ T \leq 1 \quad (6)$$

Using the k eigenvectors $\{\mathbf{e}_i \mid i = 1, 2, ..., k\}$ (where $\mathbf{e}_i$ is a $3N \times 1$ vector) obtained by using the process above; each cell image can be projected on to the eigenspace composed by matrix $\mathbf{Q}$ by projecting each matrix $\mathbf{X}_a$. And the projection of each cell image can be described as a $M \times k$ matrix $\mathbf{G}$.

$$\mathbf{G} = \mathbf{X}_a \times \mathbf{V} \quad where\ \ \mathbf{V} = \left[\begin{array}{cccc} \mathbf{e}_1 & \mathbf{e}_2 & \ldots & \mathbf{e}_k \end{array} \right] \quad (7)$$

To put it concisely, the input color image sequence is converted to a set of cell image sequences, and each sequence of cell images is stored as the matrix $\mathbf{V}$, which is the subset of eigenvectors of $\mathbf{Q}$, and the matrix $\mathbf{G}$, which is the projection onto the eigenspace. As we described in Eq.1, each sequence of cell images corresponds to one $M \times 3N$ matrix $\mathbf{X}$, and is stored as $3N \times k$ matrix $\mathbf{V}$ and $M \times k$ matrix $\mathbf{G}$, so that the compression ratio becomes that described in Eq.8.

$$compression\ \ ratio = k\frac{M + 3N}{3MN} \quad (8)$$

Each synthesized cell image can be computed by Eq.9. A virtual object image of one particular pose (pose number m) can be synthesized by aligning each corresponding cell appearance (R_m) to the 3D model.

$$\mathbf{R}_m = \sum_{i=1}^{k} g_{m,i}\mathbf{e}^T_i + [E\ \ E\ \ldots\ E] \quad (9)$$

3 Implementation

We have implemented the system described in the previous section, and have applied the *Eigen-Texture method* to real objects.

3.1 System Setup

We built a experimental system to capture the input color images and range images. In our capturing system setup, the object is attached to a rotary table. A single point light source fixed in the world coordinate is used to illuminate the object. A range image is taken through the 3 CCD color camera under a nematic liquid crystal mask[10]. A sequence of range images is taken by rotating the object; 30° by each step in this experiment. After the range images are converted into triangular mesh models[14, 15], they are merged and simplified to compose a triangular mesh model which represents the 3D shape of the object. A sequence of color images is also taken by the same 3 CCD color camera, rotating the object likewise the range image sequence, but the rotation interval is smaller than that of the range image sequence. For instance, a step of 3° was used for the first experiment describe in the next section.

3.2 Dimensions of Eigenspace

Determining the number of dimensions of the eigenspace in which the sequence of cell images are stored is a nontrivial issue, as it has significant influence on the quality of the synthesized images. According to the theory of photometric stereo[16] and Shashua's linear theory[13], three dimensions are enough for compressing and synthesizing the appearance of an object with a Lambertian surface. However, as shown in Figure 3, the images synthesized by using the data stored in only 3 dimensional eigenspace for every cell have intolerabel blurred effect around the highlights, and the textures are matted. As the reflection of most general real objects cannot be approximated by simple Lambertian reflection model due to nonlinear factors such as specular reflection and self shadow, three dimensional eigenspace

	Cat	Bear	Duck
Number of dimensions	6.56	6.96	17.6
Compression ratio	15.3:1	14.5:1	5.72:1
Error per pixel	1.46	1.79	1.79

Table 1: The results.

is not appropriate to store all the appearance change of each cell.

Also the quality of the geometric model has serious influence on the neccesary number of dimensions of eigenspace to synthesize the image precisely. The simpler the construction of geometric model, the higher the eigenspace dimensions are needed, since the triangle patches get far from approximating the object's real surface, and the correlation of each cell image becomes low. So the number of dimensions of eigenspace should differ for each cell, according to whether they have highlights or self shadows in thier sequences, and to the size of their triangle patch.

With regard to these points, we determined the number of dimensions of the eigenspace independently for each cell so that each cell could be synthesized precisely. We used eigenratio to determine the number of dimensions for each cell. For each sequence of cell images, we computed the eigenratio with Eq.6, and used the first k eigenvectors whose corresponding eigenvalues satisfied a predetermined threshold of the eigenratio. The number of dimensions for each cell required to compress the sequence of input images and to reconstruct the synthesized images can be optimized by using these cell-adaptive dimensions. Yet, on the whole, the size of the database can be reduced. This cell-adaptive dimension method can be implemented because our method deals with the sequence of input images as small segmented cell images.

Figure 4 shows the images synthesized by using 0.999 as the threshold of the eigenratio for each cell. As can be seen in Figure 4, the results described in the right side are indistinguishable from the input images shown in the left side. The number of dimensions of eigenspace used; the compression ratio and the average error per pixel are summarized in Table 1. Due to the small protudent wool around the duck's stomach, tiny self shadows appear throughout the input sequence of color images. This caused the increase of the neccesary number of dimensions for the duck compared to those necessary for the other objects.

3.3 Interpolation in eigenspace

Once the input color images are decomposed into a set of sequences of cell images, and projected onto thier own eigenspaces, interpolation between each input image can be accomplished in these eigenspaces.

As an experiment, we took thirty images of a real object as the input image sequence by rotating the object at 12° by step. By interpolating the projections of these input images in the eigenspace, we obtained interpolated projections for 3° degrees rotation by each step. By synthesizing images using these interpolated projections, images of the object whose pose were not captured in the inpute color images can be synthesized.

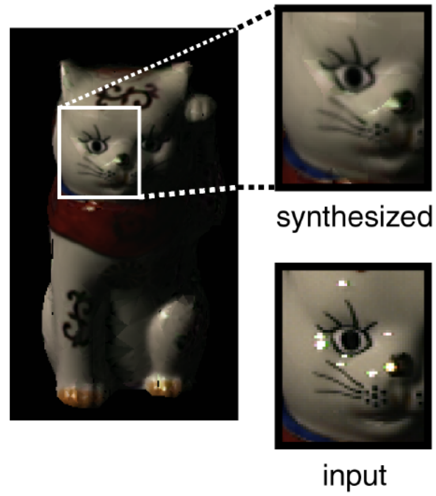

Figure 3: Virtual object images synthesized by using 3 dimensional eigenspaces.

Figure 5 shows the synthesized images with interpolation in eigenspace. The results prove that we can reconstruct synthetic images by interpolation in eigenspace with adequate accuracy. The average dimensions of eigenspace for all cells used to fill 99.9% in eigenratio was 5.1, and the compression ratio was about 20:1. But as we used only 30 images as the input image sequence and synthesized 120 images, the substantial compression ratio became almost 80:1. The average error per pixel at this time was about 7.8.

When using MPEGI to compress all the input color images into a single image sequence, the compression ratio becomes around 127 : 1 Despite this high compression ratio achieved by MPEGI, the average error per pixel becomes around 7 to 8. As image sequence compression methods such as MPEGI compress the images in their 2D coordinates without any geometric information of the target object, the errors tend to appear around the nonlinear changes of the pixel values in the images; i.e., edges of the occluding boundaries, edges of the texture, highlights, etc. These errors can not be seen in the images synthesized with *Eigen-Texture method*. Even when compressing the input color images putting the priority on the quality by using MPEGII, it is hard to avoid these errors while keeping the compression

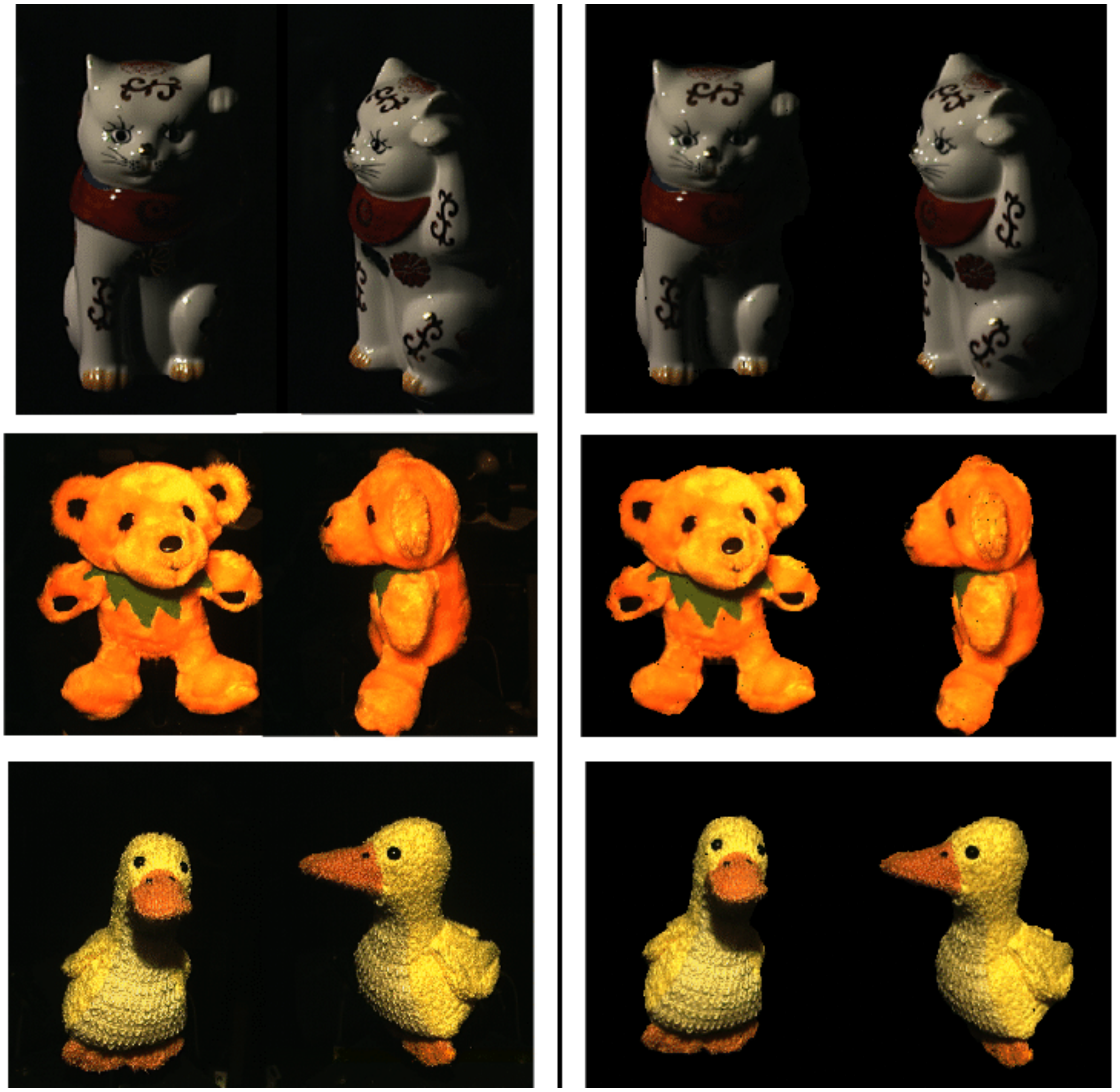

Figure 4: Left: Input color images, Right: Synthesized images (by using cell-adaptive dimensional eigenspaces).

ratio lower than *Eigen-Texture method*. With regard to these results, it can be said that the *Eigen-Texture method* is an efficient method even in terms of compression.

4 Integrating into real scene

Since our method holds an accurate 3D object model which is constructed from the range image sequence, the synthesized virtual images can be seamlessly integrated into real scene images, taking the real illumination environment into account. As shown in Eq.10, the irradiance at one point on the surface of an object from the whole illumination environment is a linear combination of the irradiance due to each light source composing the illumination environment. For that reason, a virtual object image taking into account the real illumination environment can be synthesized by decomposing the real illumination environment into several point light sources and then sampling the color images under each point light source separately .

$$\begin{aligned} I &= \int\int\{\sum_{j=1}^{n} L_j(\theta_i,\phi_i)\}R(\theta_i,\phi_i,\theta_e,\phi_e)cos\theta_i d\theta_i d\phi_i \\ &= \sum_{j=1}^{n}\int\int L_j(\theta_i,\phi_i)R(\theta_i,\phi_i,\theta_e,\phi_e)cos\theta_i d\theta_i d\phi_i \quad (10) \end{aligned}$$

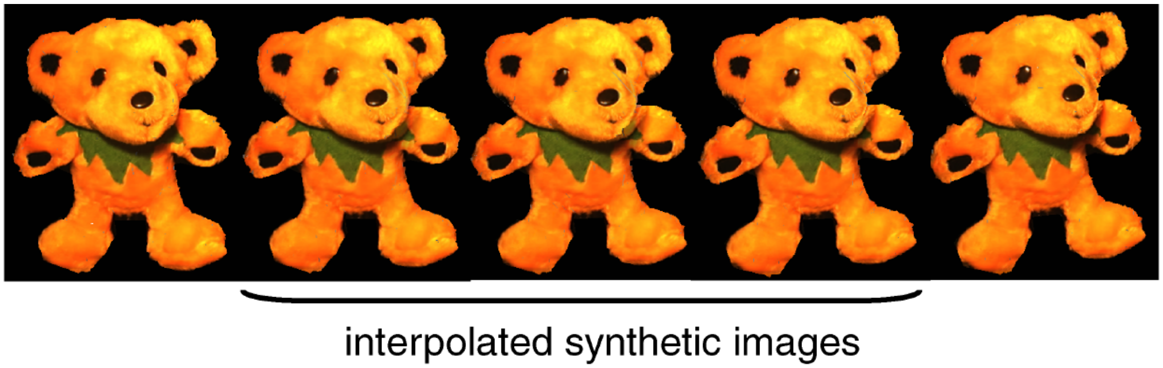

Figure 5: Virtual images reconstructed by interpolating input images in eigenspace.

where I: irradiance at one point on the object surface
L_j: radiance of one point light source
R: BRDF of the object surface
(θ_i, ϕ_i): illumination direction
(θ_e, ϕ_e): view direction

As a preliminary experiment, we took input color images under 3 point light sources lighting them separately, and synthesized images of the virtual object with all lights turned on. Under each point light source, we took 40 color images and synthesized 120 virtual object images for each light source with the threshold 0.999 in eigenratio with interpolation in eigenspace; we then synthesized the virtual object image sequence with all lights on by taking a linear combination of each point light source virtual object image. Figure 6 shows the result of the linear combination of a specific pose. In addition, we integrated the virtual object image sequence into a real scene image, which also had been taken under a condition in which all lights were on. The result of the integration, which is shown in Figure 7, proves that our method enables the creation of the accurate appearance of the object surface and the precise shadow of the virtual object according to the real illumination environment[9, 2].

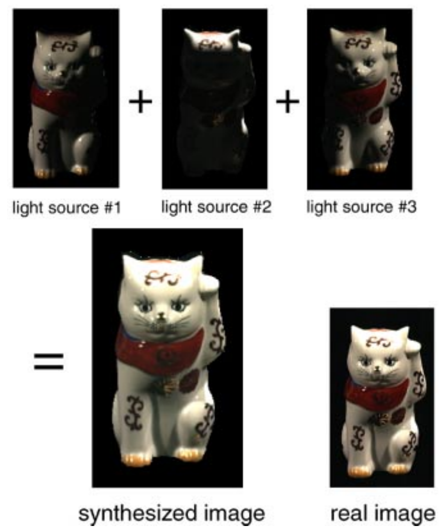

Figure 6: Linear combination of light sources.

5 Conclusions

We have proposed the *Eigen-Texture method* as a new rendering method for synthesizing virtual images of an object from a sequence of range and color images. The implementation of the method proves its effectiveness, in particular, its high compression ratio. Also, we have demonstrated seamless integration of virtual appearance with real background images by using the method. The merits of the proposed method can be summarized as follows.

First, high compression ratio can be achieved, because we compress a sequence of cell images corresponding to a physical patch on the object surface. Appearance variation in the sequence is approximately limited to brightness change due to illumination geometry. Second, interpolation in eigenspace can achieve synthesization of a virtual object image when object pose is not included in the input image sequence. Owing to this interpolation in eigenspace, we can reduce the necessary number of sampling color images, and reduce the amount of data to be stored. Third, a wide range in application is possible, because we do not need any detailed reflectance analysis. The method can be applied to such objects as those with rough surfaces, i.e., the bear and duck shown in Figure 4, or with strong highlights, and whose color values saturate the dynamic range of a CCD camera, such as the cat in Figure 4. Lastly, as an accurate 3D model of the real object is constructed from the input range images, we can generate accurate cast shadows in integrating virtual images with real background im-

Figure 7: Integrating virtual object into real scence.

ages. Thanks to the linearity in image brightness, we can decompose the real illumination distribution into separate point light sources, and sample the object under each point light source separately.

On the other hand, our method has a few drawbacks. In particular, the computational expense for compression using eigenspace could be large. As solving the eigenstructure decompostion problem requires a large number of iterations, the computation cost for storing the input image sequence in eigenspace becomes relatively expensive, although, once the input images are compressed and stored in eigenspace, the synthesis of virtual images can be computed in real time. With regard to this point, we beleive our method can take advantage especially on applications for interactive mixed reality systems, such as virtual museums and electric shopping malls, where the targets can be compressed beforehand off line.

References

[1] S.E. Chen and L. Williams. View interpolation for image synthesis. In *Computer Graphics Proceedings, ACM SIGGRAPH 93*, pages 279–288, Aug. 1993.

[2] P. Debevec. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *Computer Graphics Proceedings, ACM SIGGRAPH 98*, pages 189–198, Jul. 1998.

[3] S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen. The Lumigraph. In *Computer Graphics Proceedings, ACM SIGGRAPH 96*, pages 43–54, Aug. 1996.

[4] M. Levoy and P. Hanrahan. Light Field Rendering. In *Computer Graphics Proceedings, ACM SIGGRAPH 96*, pages 31–42, Aug. 1996.

[5] A. Matsui, K. Sato, and K. Chihara. Composition of Real Light and Virtual Light based on Environment Observation and KL-expansion of Multiple Lightsource Image. *Technical Report of IEICE PRMU97-115*, 97(324):29–36, Oct. 1997.

[6] H. Murase and S.K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[7] K. Nishino, I. Sato, Y. Sato, and K. Ikeuchi. Eigen-Texture Method: Appearance Compression based on 3D Model. In *IPSJ Symposium Series, MIRU'98(in Japanese)*, volume 98, pages I–19–I–26, 7 1998.

[8] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle. View-based Rendering: Visualizing Real Objects from Scanned Range and Color Data. In *Proceedings of 8th Eurographics Workshop on Rendering*, Jun. 1997.

[9] I. Sato, Y. Sato, and K. Ikeuchi. Seamless integration of virtual objects into a real scene based on a real illumination distribution. Technical Report IIS-CVL-98-101, Computer Vision Lab., IIS, The Univ. of Tokyo, Jan. 1998.

[10] K. Sato and S. Inokuchi. Range-imaging system utilizing nematic liquid crystal mask. In *First International Conference on Computer Vision, IEEE*, pages 657–661, 1987.

[11] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. *Journal of the Optical Society of America*, 11(11):2990–3002, 1994.

[12] Y. Sato, M.D. Wheeler, and K. Ikeuchi. Object shape and reflectance modeling from observation. In *Computer Graphics Proceedings, ACM SIGGRAPH 97*, pages 379–387, Aug. 1997.

[13] A. Shashua. *Geometry and Photometry in 3D Visual Recognition*. PhD thesis, Dept. Brain and Cognitive Science, MIT, 1992.

[14] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Computer Graphics Proceedings, ACM SIGGRAPH 94*, pages 311–318, Jul. 1994.

[15] M.D. Wheeler, Y. Sato, and K. Ikeuchi. Consensus Surfaces for Modeling 3D Objects from Multiple Range Images. In *Sixth International Conference on Computer Vision, IEEE*, pages 917–924, 1998.

[16] R.J. Woodham. Photometric Stereo: A Reflectance Map Technique for Determining Surface Orientation from a Single View. In *Image Understanding Systems and Industrial Applications*, volume 155, pages 136–143. Proc. SPIE 22nd Annual Technical Symp., Aug. 1978.

[17] Z. Zhang. Modeling Geometric Structure and Illumination Variation of a Scene from Real Images. In *Sixth International Conference on Computer Vision*, pages 1041–1046, Jul. 1998.