Appearance Compression and Synthesis based on 3D Model for Mixed Reality

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

Rendering photorealistic virtual objects from their real images is one of the main research issues in mixed reality systems. We previously proposed the Eigen-Texture method, a new rendering method for generating virtual images of objects from thier real images to deal with the problems posed by past work in image-based methods and model-based methods. Eigen-Texture method samples apperances of a real object under various illumination and viewing conditions, and compresses them in the 2D coordinate system defined on the 3D model surface. However, we had a serious limitation in our system, due to the alignment problem of the 3D model and color images. In this paper we deal with this limitation by solving the alignment problem; we do this by using the method orginally designed by Viola[14]. This paper describes the method, and reports on how we implement it.

1. Introduction

Recently, mixed reality systems have been one of the main topics in the research areas of computer vision and computer graphics. Main research issues in these mixed reality systems include how to render photorealistic virutal objects from their real images, and then how to seamlessly integrate those virtual objects with real images. Our interest is on the former topic: rendering virtual object images from real images. The CV and CG communities have proposed two representative rendering methods to obtain such virtual images from real objects: image-based and model-based methods.

The basic idea of representative image-based methods[1, 2, 3, 5], is to acquire a set of color images of a real object and store them on the disk of a computer efficiently (via compression), and then synthesize virtual object images either by selecting an appropriate image from the stored set or by interpolating multiple images. Since the main purpose of image-based rendering is to render virtual images simply from real images without analyzing any reflectance characteristics of 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. To avoid making the system complicated, few image-based rendering methods employ accurate 3D models of real objects. Although Lumigraph[3] uses a volume model of the object for determining the basis function of lumigraphs, the volume model obtained from 2D images is not accurate enough to be used in mixed reality systems, i.e. for the purpose of making cast shadows under real illuminations corresponding to the real background image.

Unlike image-based methods, model-based methods[10, 11] analyze the reflectance characteristics on the surface of the real object to render photorealistic virtual objects by assuming reflectance models. Since the reflectance parameters are obtained at every surface point of the object, integration of synthesized images with real background can be accomplished quite realistically; the method can generate a realistic appearance of an object as well as shadows cast by the object onto the background. However, model-based rendering has nontrivial intrinsic constraints; it cannot be applied to objects whose reflectance properties cannot be approximated by using simple reflectance models. Furthermore, since computation of reflectance parameters needs the surface normal at each point of the surface, it cannot be applied to objects whose surfaces are rough.

To overcome the problems posed by the previous methods, we previously proposed a new rendering method, which we refer to as the *Eigen-Texture method*[7]. 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

0-7695-0164-8/99 $10.00 (c) 1999 IEEE

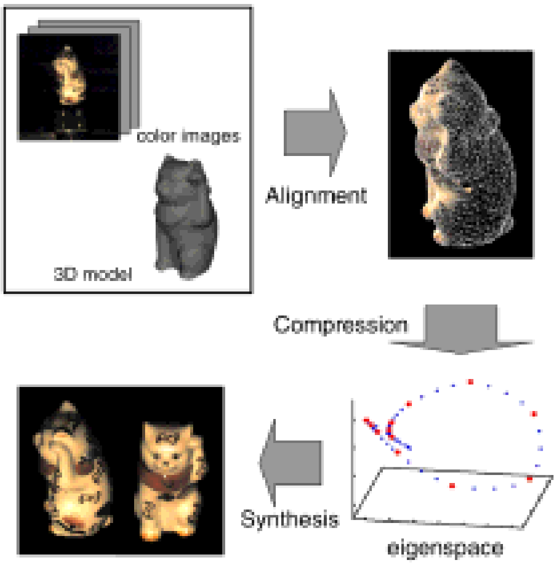

Figure 1. Outline of the Eigen-Texture method.

coordinate system defined on the 3D model surface. Cast shadows can be generated using the 3D model. Because the method does not assume any reflectance model as model-based methods do, it can be applied to a wide variety of objects. Also, thanks to the linearity of image brightness, virtual images under a complicated illumination condition can be generated by summation of component virtual images sampled under single illuminations.

However, we experienced a severe constraint in our system flexibility. Since we used a light stripe range finder[4, 9] to obtain the range image sequence, the range images and the color images were acquired in the same coordinate, and the alignment problem of the 3D model and the color images could be disregarded. In this paper, we describe how we use a laser range finder to obtain the range images, and we solve the alignment problem of the 3D model and color images by using the method originally designed by Viola[14]: alignment by maximization of mutual information.

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 method. In Section 4 we describe our future work and conclude this paper.

2. Eigen-Texture Method

In this section we describe the theory of the *Eigen-Texture method*.

Figure 1 displays an overview of the proposed method. The *Eigen-Texture method* takes both intensity images and range images as the input, and a 3D model of an object is created from a sequence of range images with mesh generation from each range image, registration and integration of each mesh models, and mesh decimation of the final 3D model [13, 15]. Usually the intensity images and the range images are taken in different coordinates, unless we use a light stripe range finder to acquire both image sequences with the same CCD camera. The first step of our method is to align each intensity image to the 3D model. We accomplish this alignment by using the method originally designed by Viola; this method aligns 3D model and intensity image by maximizing the mutual information between them(2.1). Once each intensity image is aligned, the method pastes each intensity 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.

2.1. Alignment

After generating an accurate 3D model of an object, the *Eigen-Texture method* aligns each input color image with the 3D model. This alignment can be accomplished by using the method originally designed by Viola, maximizing the mutual information between the 3D model and each color image. Given a 3D model of an object and a pose, a model for the imaging process, such as reflectance models, could be used to predict the image that will result. For example, if the object has an Lambertian surface, the intensity image can be predicted from surface normals of the object surface. The predicted image can then be directly compared to the actual image. If the object model and pose are correct, the predicted and actual images should be identical, or close to it. The alignment method proposed by Viola is based on a formulation of the mutual information between the model and the image to predict the image from the model.

Let us define $u(x)$ the property of the model at a particular point x on the model surface, and $v(T(x))$ the property of the image at the point corresponding to x with transformation T. The mutual information between the model and the image can be described as Eq.1.

$$I(u(x), v(T(x))) \equiv h(u(x)) + h(v(T(x))) - h(u(x), v(T(x))) \quad (1)$$

The first term in Eq.1 is the entropy in the model, and is not a function of T. The second term is the entropy of the part of the image into which the model projects. It encourages transformatins that project the model u into complex parts of the image v. The third term is the negative joint entropy the model and the image, and it encourages transformations where the model explains the image well. The alignment problem of the model and the images can be replaced by finding an estimate of the transformation T that aligns the model and image by maximizing their mutual information over the transformations T.

$$\hat{T} = \arg\max_{T} I(u(x), v(T(x))) \quad (2)$$

To solve this problem, Viola's method first estimates the entropies from samples: approximates the underlying probability density $p(z)$ by a superposition of Gaussian densities centered on the elements of a sample A drawn from z, Eq.3, and approximate statistical expectation with the sample average over another sample B drawn from z, Eq.4.

$$\begin{aligned} p(z) &\approx \frac{1}{N_A}\sum_{z_j\in A} G_\psi(z - z_j) && (3) \\ E_z(f(z)) &\approx \frac{1}{N_B}\sum_{z_i\in B} f(z_i) && (4) \\ where\ G_\psi(z) &\equiv (2\pi)^{\frac{-n}{2}}|\psi|^{\frac{-1}{2}}\exp(-\frac{1}{2}z^T\psi^{-1}z) \end{aligned}$$

With these approximations, the entropy of a random variable z can be approximated as follows:

$$\begin{aligned} h(z) &= -E_z(\ln p(z)) \\ &\approx \frac{-1}{N_B}\sum_{z_i\in B}\ln\frac{1}{N_A}\sum_{z_j\in A} G_\psi(z_i - z_j) && (5) \end{aligned}$$

This approximation of entropy may now be used to approximate the derivative of the mutual information as follows:

$$\begin{aligned} \widehat{\frac{dI}{dT}} &= \frac{d}{dT}h(v(T(x))) - \frac{d}{dT}h(u(x), v(T(x))) \\ &= \frac{1}{N_B}\sum_{x_i\in B}\sum_{x_i\in A}(v_i - v_j)^T[W_v(v_i, v_j)\psi_v^{-1} \\ &- W_{uv}(w_i, w_j)\psi_{vv}^{-1}]\frac{d}{dT}(v_i - v_j) && (6) \end{aligned}$$

$$\begin{aligned} where\ W_v(v_i, v_j) &\equiv \frac{G_{\psi_v}(v_i - v_j)}{\sum_{x_k\in A} G_{\psi_v}(v_i - v_k)} \\ W_{uv}(w_i, w_j) &\equiv \frac{G_{\psi_{uv}}(w_i - w_j)}{\sum_{x_k\in A} G_{\psi_{uv}}(w_i - w_k)} \end{aligned}$$

$$u_i \equiv u(x_i), v_i \equiv v(T(x_i)), w_i \equiv [u_i, v_i]^T$$

$$\psi_{uv}^{-1} = DIAG(\psi_{uu}^{-1}, \psi_{vv}^{-1}) \quad (7)$$

The covariance matrices of the component densities used in the approximation scheme for the joint density are block diagonals(Eq.7). The local maximum of mutual information is sought taking repeated steps that are proportional to the approximation of the derivative of the mutual information with respect to the transformation as follows. The procedure is repeated a fixed number of times or until convergence is detected.

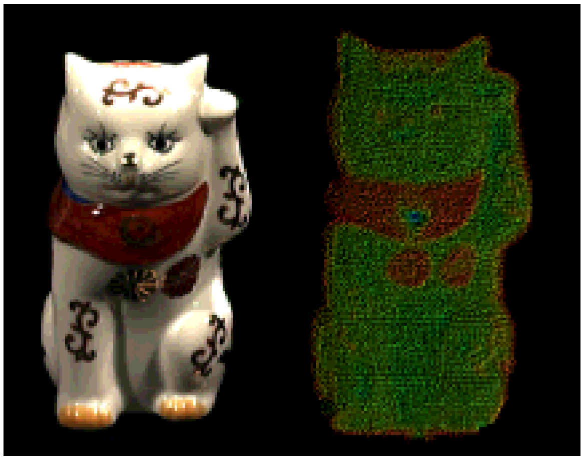

Figure 2. A color image and a range image with the reflectance power attribute.

```
$Repeat:$
$A \leftarrow \{sample\ of\ size\ N_A\ drawn\ from\ x\}$
$B \leftarrow \{sample\ of\ size\ N_B\ drawn\ from\ x\}$
$T \leftarrow T + \lambda\frac{\widehat{dI}}{dT}$
```

In the experiments described in 3, we used a laser range finder to acquire the range image sequence. Likewise most other laser range finders, the laser range finder we used (PULSTEC TDS-1500) returns the reflectance power value of the laser at each point the laser hits on the object surface as shown in Figure 2[1]. This value depends predominantly on the texture of the model surface, so that it is highly correlative with the intensity values of the object surface in their color images. Regarding this fact, we used these reflectance power values as the property of the model to be evaluated in the alignment procedure, as well as the surface normal at each point of the model surface. The pixel values are evaluated as the property of the image.

2.2. Appearance Compression and Synthesis

After the alignment is accomplished, each color image is divided into small areas that correspond to triangular patches on the 3D model. Each triangular patch is normalized to have the same shape and size as that of the others.

[1] A clear image can be found in the color version of this paper put on the CD-ROM of the proceedings.

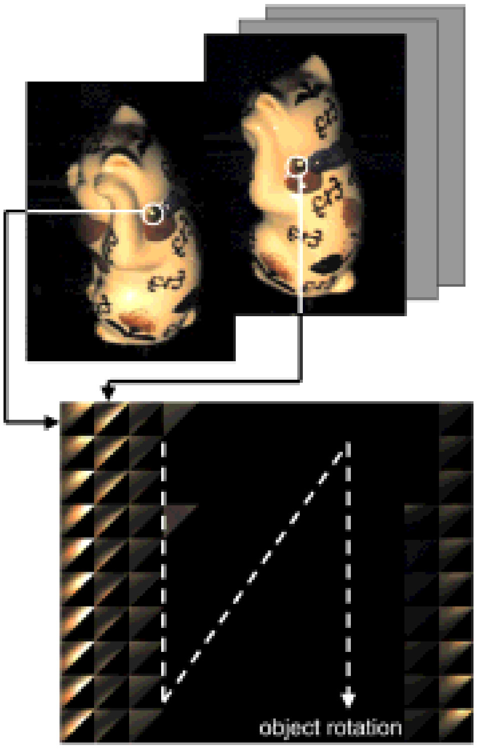

Figure 3. A sequence of cell images.

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 3. 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.8). 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 the pose number.

$$\mathbf{X}_m = \left[\begin{array}{ccccc} x^Y_{m,1} & ... & x^{C_r}_{m,1} & ... & x^{C_b}_{m,N} \end{array} \right] \quad (8)$$

Then the sequence of cell images can be represented as a $M \times 3N$ matrix as shown in Eq.9.

$$\mathbf{X} = \left[\begin{array}{cccc} \mathbf{X}^T_1 & \mathbf{X}^T_2 & ... & \mathbf{X}^T_M \end{array} \right]^T \quad (9)$$

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.

$$\mathbf{X}_a = \mathbf{X} - \left[\begin{array}{ccc} E & ... & E \\ . & ... & . \\ E & ... & E \end{array} \right] \quad (10)$$

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

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.

$$\mathbf{Q} = \mathbf{X}^T_a \mathbf{X}_a \quad (11)$$

$$\lambda_i \mathbf{e}_i = \mathbf{Q}\mathbf{e}_i \quad (12)$$

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.13).

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{3N} \lambda_i} \geq T \quad where \;\; T \leq 1 \quad (13)$$

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 $\mathbf{G}$.

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

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

0-7695-0164-8/99 $10.00 (c) 1999 IEEE

of eigenvectors of $\mathbf{Q}$, and the matrix $\mathbf{G}$, which is the projection onto the eigenspace. As we described in Eq.8, 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.14.

$$compression\ \ ratio = k\frac{M+3N}{3MN} \tag{14}$$

Each synthesized cell image can be computed by Eq.15. 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.

$$\mathbf{R}_m \ = \ \sum_{i=1}^{k} g_{m,i}\mathbf{e}_i^T + [E\ \ E\ \ ...\ \ E] \tag{15}$$

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

For the experiments, we attach the object to a rotary table, and use light sources fixed in the world coordinate to illuminate the object. A range image is taken through a laser range finder PULSTEC TDS-1500 . A sequence of range images is taken by incrementing the rotation angle of the rotary table for each step; 30° by each step in this experiment. After the range images are converted into triangular mesh models[13, 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 taken by a SONY 3 CCD color camera, incrementing the rotation angle as well as 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. Cell-adaptive Dimensional 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 trilinear theory[12], three dimensions are enough for compressing and synthesizing the appearance of an object with a Lambertian surface. However, 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 is not appropriate to store all the appearance change of each cell.

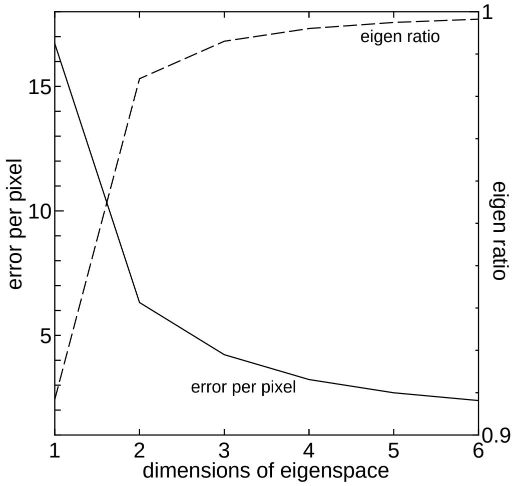

Figure 4. Graph of the eigen ratio and the error per pixel versus the number of dimensions of eigenspace.

Also the quality of the geometric model has serious inpact on the number of dimensions of eigenspace neccesary to precisely synthesize the image. The simpler the construction of the geometric model, the higher the eigenspace dimensions are needed, since the trianglular patches stray far from a close approximation of 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 triangular patch.

With regard to these points, we determined the number of dimensions of the eigenspace independently for each cell so that each could be synthesized precisely. We used eigenratio to determine the number of dimensions for each cell. Figure 4 describes the relation between the eigen ratio, the error per pixel versus the number of dimensions of the eigenspace. Here the eigen ratio is the ratio of the eigen values in the whole summation of them, and the error per pixel describes the average difference of the pixel values in 256 graduation in the synthesized and real image. The eigen ratio is in inverse proportion to the error per pixel, so that it is reasonable to threshold the eigen ratio to determine the number of dimsensions of the eigenspace. For each sequence of cell images, we computed the eigenratio with Eq.13, 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 re-

	Cat	Raccoon
Average number of dimensions	6.56	18.2
Compression ratio	15.3:1	5.54:1
Essential compression ratio	8.34:1	3.32:1
(MB)	(357:42)	(319:96)
Average error per pixel	1.46	1.34

Table 1. The number of dimensions, the compression ratio and the average error per pixel of the results.

construct 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 with small segmented cell images.

Figure 5 shows the images synthesized by using 0.999 as the threshold of the eigenratio for each cell. As can be seen in Figure 5, the results described in the right side are indistinguishable from the input images shown in the left side. The average number of dimensions of eigenspace used, the compression ratio and the average error per pixel are summarized in Table 1. Because our method does not assume any reflectance model and does not have to analyze the surface properties, it can be applied to objects with rough surface like the raccoon in Figure 5, which is difficult with model-based methods.

3.3. Compression Ratio

The compression ratio defined in Eq.14 is the compression ratio of the cell image sequences, which is computed from the number of dimensions, the size of each cell image and the number of input color images. The essential compression ratio described in Table 1 is also the compression ratio between the cell image sequence and the data size to be stored in eigenspaces, but its value is computed from the real data size on the computer. The values of this essential compression ratio is relatively lower than the compression ratio computed by Eq.14. This is caused by the difference of data types: the cell image sequences derived from the input color image sequence are represented with $unsigned\ char$ type, while the data to be stored in eigenspaces, the projections and eigenvectors, are represented with $float$ type. To avoid this loss in compression due to data types, we are working on implemention of vector quantization of compressed stored data.

In this paper, we are not computing the compression ratio between the input image sequence and the data size stored in eigenspaces because an accurate comparison cannot be accomplished due to the resolution problem of cell images. When *Eigen-Texture method* converts the input image sequence into a set of cell image sequences, the resolution of the cell images are determined to a fixed number with regard to the largest triangular patch on the 3D model surface. Although this enables us to synthesize virtual images in higher resolution compared to the input images, it causes loss in compression ratio. We are now investigating a method to determine the resolution of cell images adaptively to thier corresponding triangular patch size. Furthermore, to derive higher compression ratio, mesh generation of the 3D model with regards to the color attributes on its surface is under our consideration.

3.4. 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. The square points illustrated in Figure 6 indicate projections of each cell image in eigenspace corresponding to each pose of the real object. By interpolating these projected points in the eigenspace, we obtained interpolated projection points for each position of the real object for 3° degrees rotation by each step. As a practical matter, we interpolated the projection of the input image sequence for each eigenvector and obtained the interpolated projection points denoted by plus marks in Figure 6. By synthesizing images using these interpolated projections, object images whose pose were not captured in the input color images could be synthesized. It is impossible to interpolate the appearance features, i.e. highlights, that are not captured in the input color image sequence, so that synthetic images interpolated from very sparse input sequence tend to lack those photorealistic characterisitcs. Taking this into account, typical objects with shiny surfaces like the cat in Figure 5 could be compressed about 80:1 with interpolation in eigenspaces; the synthesized results would be satisisfactory realistic.

4. Conclusions and Future Work

In this paper, we have proposed the *Eigen-Texture method* as a rendering method for synthesizing virtual images of an object from a sequence of range and color images, along with improvement on its constraint on the image acquiring system. To accomplish the alignment procedure of the 3D model and color images, we use the method originally designed by Viola, alignment of 3D model and color images by maximizing the mutual information. As recent ordinary laser range finders return the power of laser

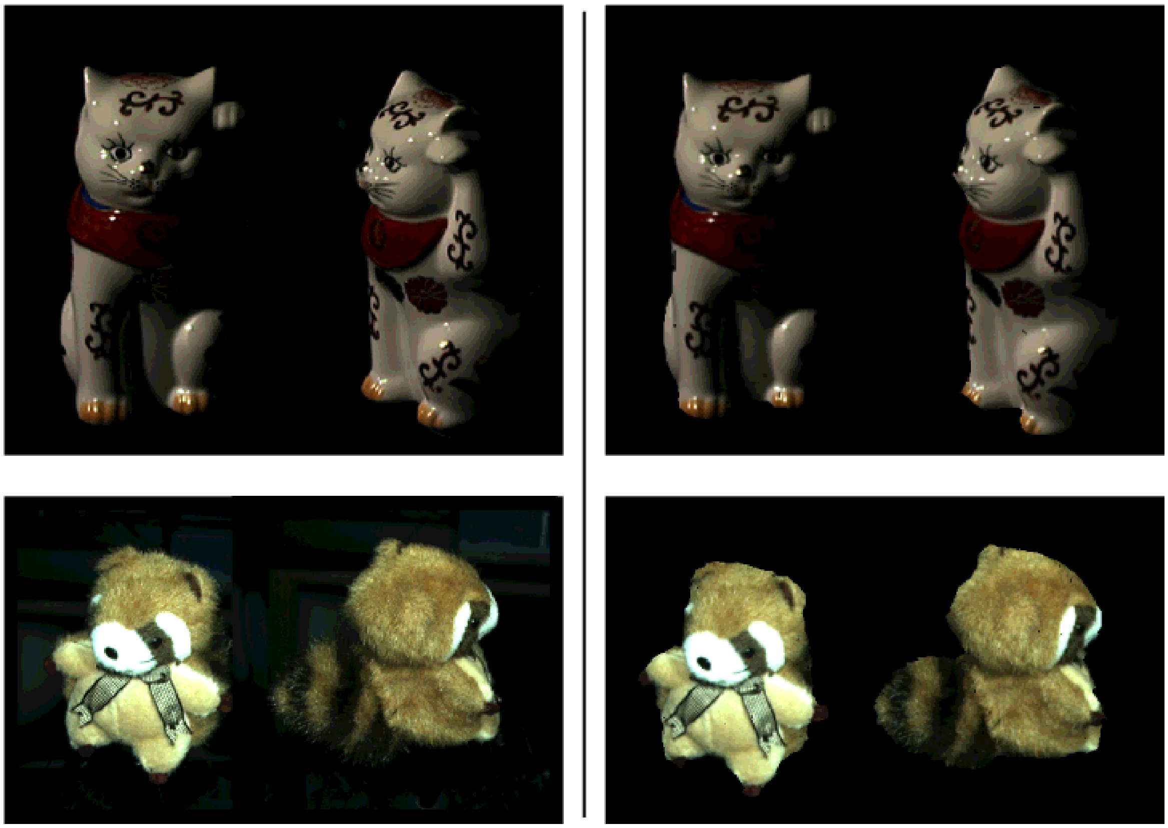

Figure 5. Left: Input color images, Right: Synthesized images (by using cell-adaptive dimensional eigenspaces).

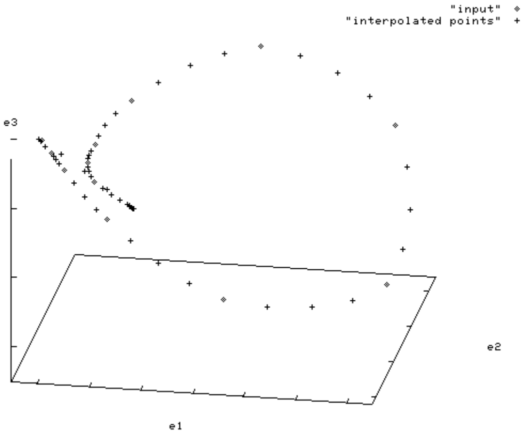

Figure 6. G plotted in eigenspace (the first three eigenvectors are depicted).

reflectance at each point scanned by the laser, we use this information to raise the robustness of the alignment. With this improvement on the alignment problem, our method has gained flexibility in application as well as the ability to be applied to a wide variety of objects and high compression ratio.

Figure 7 shows an intensity image and two range images with the reflectance power attribute of the 13m tall buddha sitting in the city of Kamakura, Japan. By using recent products in the field of laser range finders [2], it is becoming relatively easy to obtain accurate range data of large statues and buildings. As a practical application of our method, we are planning to a scan heritage objects in Japan and Asia, such as the buddha in Kamakura city, and thereby archive the geometric and photometric properties of them for the preservation of historic resources.

[2] We acquired the range images of the buddha by a time-of-flight laser range scanner Cyrax2400, a product of Cyra Tech. Inc.

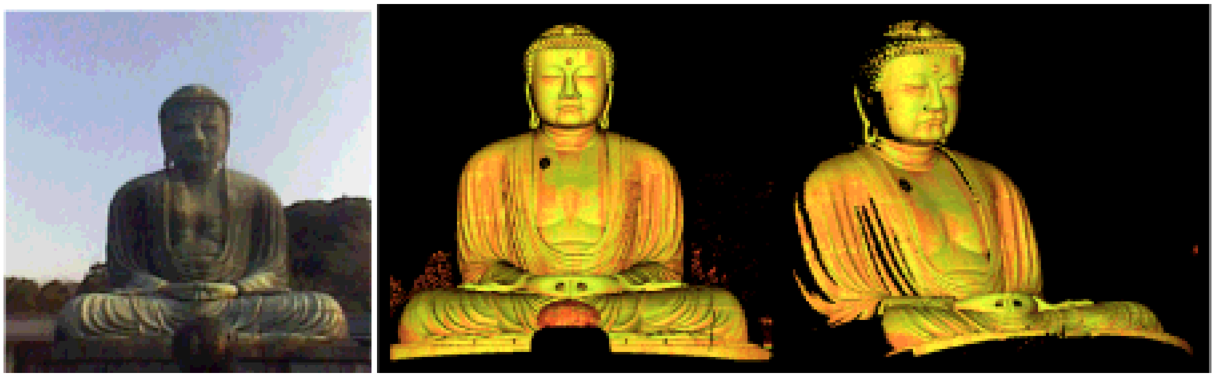

Figure 7. A color image and range image with reflectance attribute of the huge buddha in Kamakura city.

5. Acknowledgements

Special thanks to CAD Center Corporation and Koutokuin for their help in scanning the huge buddha. This work was partially supported by the Ministry of Education, Science, Sports and Cultrure, Grant-in-Aid for Scientific Research (B), 09450164, 1999.

References

[1] S. Chen. Quicktime VR: An image-based approach to virtual environment navigation. In *Computer Graphics Proceedings, ACM SIGGRAPH 95*, pages 29–38, Aug. 1995.

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

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

[4] K. Ikeuchi and K. Sato. Determining Reflectance Properties of an Object Using Range and Brightness Images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(11):1139–1153, Nov. 1991.

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

[6] 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.

[7] K. Nishino, Y. Sato, and K. Ikeuchi. Eigen-Texture Method: Appearance Compression based on 3D Model. In *Proc. of Computer Vision and Pattern Recognition '99*, volume 1, pages 618–624, Jun. 1999.

[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] 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.

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

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

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

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

[14] P. Viola. *Alignment by Maximization of Mutual Information*. PhD thesis, MIT AI Lab., Jun. 1995.

[15] M. 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. 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.