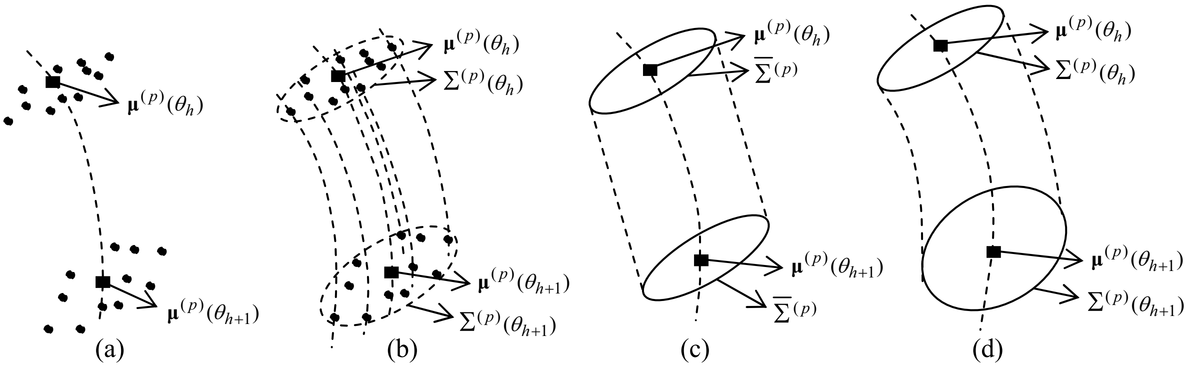

Figure 1. Construction models of appearance manifold (a) Parametric Eigenspace (PE), (b) points interpolation (AMPI), (c) constant covariance matrix (AMCC), (d) view-dependent covariance matrix (AMVC).

Generally, the captured images should be normalized in brightness and scaled in order to be invariant to image magnification and illumination intensity. These normalized images can be written as a vector $\mathbf{x}$ by reading the number of pixels (N) in an image:

$$\mathbf{x} = [x_1, x_2, ..., x_N]^T \ . \tag{1}$$

Let M be the number of images in a learning set. By subtracting the average image c of all images, the learning set $\mathbf{Y}$ will be obtained:

$$\mathbf{Y} = [\mathbf{x}_1 - c, \mathbf{x}_2 - c, ..., \mathbf{x}_M - c] \ . \tag{2}$$

Next, define the auto-correlation matrix by

$$\mathbf{Q} = \mathbf{YY}^T \tag{3}$$

and determine the eigenvalues $\lambda_\mathbf{i}$ with their corresponding eigenvectors $\mathbf{e_i}$ by solving the following eigenvector decomposition problem:

$$\lambda_\mathbf{i}\mathbf{e_i} = \mathbf{Qe_i} \ . \tag{4}$$

To reduce the dimension, simply ignore small eigenvalues and use only k corresponding eigenvectors by using a T threshold value:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i} \geq T \ , \tag{5}$$

where $k << N$.

The first k eigenvectors will be used to project S learning samples of P objects with H poses. Project $\mathbf{x}_s^{(p)}(\theta_h)$ as the s sample image of object p with horizontal viewpoint θ_h into the eigenspace:

$$\mathbf{g}_s^{(p)}(\theta_h) = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_k]^T \left(\mathbf{x}_s^{(p)}(\theta_h) - \mathbf{c}\right) \ . \tag{6}$$

By projecting all of the learning samples into the eigenspace, learning features are represented efficiently as a set of discrete points in a low-dimensional space.

2.2. Construction of Appearance Manifold with Embedded Covariance Matrix

In this section, we present various techniques to construct the appearance manifold. Figure 1 shows the four types of construction models for the appearance manifold: the simple manifold used in the Parametric Eigenspace (PE) method, the appearance manifold using the point interpolation (AMPI) method, the appearance manifold with constant covariance matrix (AMCC) method, and the appearance manifold with view-dependent covariance matrix (AMVC) method.

Although each method uses a different type of construction technique for the appearance manifold, in general they use the same basic steps. First, after transforming learning images to the eigenspace, calculate the mean vector $\mu^{(p)}(\theta_h)$ and the covariance matrix $\Sigma^{(p)}(\theta_h)$ for each object p for horizontal viewpoint θ_h . The mean vector is typically estimated using

$$\mu^{(p)}(\theta_h) = \frac{1}{S}\sum_{s=1}^{S} \mathbf{g}_s{}^{(p)}(\theta_h) \ , \tag{7}$$

where s is the number of learning samples from each class, and $\mathbf{g}_s{}^{(p)}(\theta_h)$ is the image sample s from class viewpoint θ_h and object p . On the other hand, the covariance matrix is typically estimated by

$$\Sigma^{(p)}(\theta_h) = \frac{1}{S-1}\sum_{s=1}^{S} \left(\mathbf{g}_s{}^{(p)}(\theta_h) - \mu^{(p)}(\theta_h)\right)\left(\mathbf{g}_s{}^{(p)}(\theta_h) - \mu^{(p)}(\theta_h)\right)^T \tag{8}$$

Next, create $\tilde{\mu}^{(p)}(\theta)$ as a continuous manifold of the mean vector and $\tilde{\Sigma}^{(p)}(\theta)$ for the covariance matrix. The processes of creating manifolds $\tilde{\mu}^{(p)}(\theta)$ and $\tilde{\Sigma}^{(p)}(\theta)$ might be different from one method to another.

The PE method uses a simple manifold obtained from the interpolation of the mean vector of the eigenpoints in two consecutive poses. However, for the covariance matrices, the PE method simply applies the values of the identity matrix. The construction model of the appearance manifold in the PE method is depicted in Fig. 1(a).

Next, Fig. 1(b) shows the appearance manifold with the point interpolation (AMPI) method. It obtains the appearance manifold by interpolating every eigenpoint in each pose class to other eigenpoints in the consecutive pose classes that have similar characteristics, such as degradation effects. After creating those manifolds for each eigenpoint, generate the new eigenpoints for every in-between class pose, and then calculate their mean vectors and covariance matrices for every pose class.

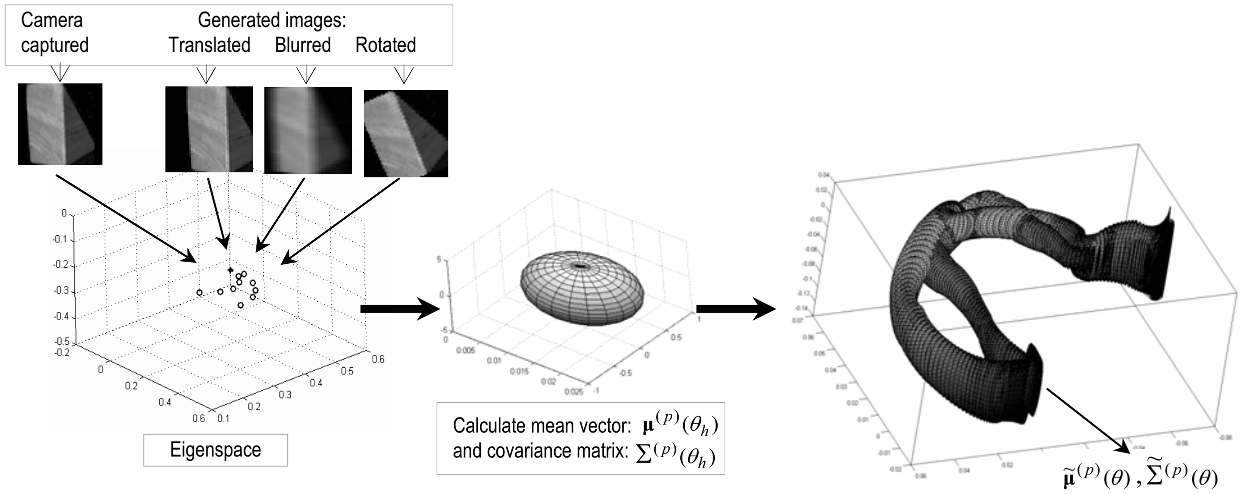

Figure 2. Scheme of AMVC method for 3D object recognition.

Figure 1(c) shows the tube appearance manifold with a constant covariance matrix (AMCC). After calculating the mean vectors and covariance matrix values for each learning pose in (7) and (8), apply an interpolation method for the mean vector of two consecutive learning poses to obtain the manifold of mean vector $\tilde{\boldsymbol{\mu}}^{(p)}(\theta)$. On the other hand, the manifold of covariance matrix $\tilde{\Sigma}^{(p)}(\theta)$ contains the same value for every viewpoint θ_h by applying the average covariance matrix

$$\bar{\Sigma}^{(p)} = \frac{1}{H}\sum_{h=1}^{H}\Sigma^{(p)}(\theta_h) \tag{9}$$

with H number of viewpoint classes for each object.

Next, Fig. 1(d) shows another type of appearance manifold method, called the appearance manifold with view-dependent covariance matrix (AMVC). This method uses the appearance manifold embedded with view-dependent covariance matrix that changes along with the function of viewpoints. The manifold $\tilde{\boldsymbol{\mu}}^{(p)}(\theta)$ could be obtained by applying an interpolation method between two consecutive mean vectors $\boldsymbol{\mu}^{(p)}(\theta_h)$ and $\boldsymbol{\mu}^{(p)}(\theta_{h+1})$. Then, the manifold $\tilde{\Sigma}^{(p)}(\theta)$ could be obtained by interpolating the covariance matrices $\Sigma^{(p)}(\theta_h)$ and $\Sigma^{(p)}(\theta_{h+1})$, respectively. Here, since we use only the horizontal pose parameter θ_h, the surface of the appearance manifold in the AMVC method will be a tube. Figure 2 shows the scheme of the AMVC method with its tube surface.

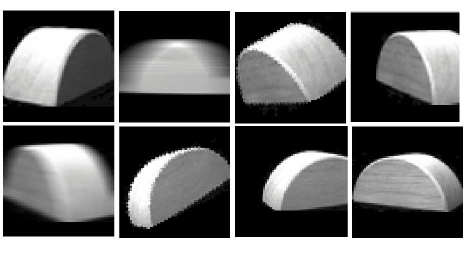

Figure 3. Sample images of 3D objects with various translation, rotation, and motion blur effects.

2.3. Classification using distance measurement

In order to recognize an input image $\mathbf{u}$, first project it into the eigenspace

$$\mathbf{z} = [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_k]^T(\mathbf{u} - \mathbf{c}) \tag{10}$$

and then calculate distance d between the projected-image in the eigenspace $\mathbf{z}$ and the manifold.

Since we have the parameter of mean vector and covariance matrix in the appearance manifold, the sufficient distance measurement to classify the input image is the Mahalanobis distance, defined in this formula:

$$d^{(p)}(\mathbf{z}) = \min_{\theta}\left(\mathbf{z} - \tilde{\boldsymbol{\mu}}^{(p)}(\theta)\right)^T\left(\tilde{\Sigma}^{(p)}(\theta)\right)^{-1}\left(\mathbf{z} - \tilde{\boldsymbol{\mu}}^{(p)}(\theta)\right) . \tag{11}$$

The Mahalanobis metric provides a sufficient way to classify images based on their related characteristics and likelihood in each pose class.

3. Application in 3D Object Recognition

To evaluate the performance of our proposed methods, explained in section 2.2, we developed an application in 3D object recognition. The developed system was used to recognize seven objects with various horizontal pose positions and influenced by geometric and quality-degradation effects, such as translation, rotation, and motion blur. Samples of 3D objects with various translation, rotation, and motion blur effects are shown in Fig. 3.

In the learning stage, the images were first normalized into 32 x 32-pixel grayscale images. Then, the system was learned with a total of 6,552 images. Each object consists of 36 poses with 10-degree intervals of horizontal positions (0^o, 10^o, 20^o…350^o), and each pose consists of 26 learning images with an original camera-captured image and 25 generated images with various degradation effects. Those generated images were obtained by composing artificial noises with the MATLAB program, such as left and right translations (3, 6, 9, 12, 15 pixels), clockwise and counter-clockwise rotations (5^o, 10^o, 15^o, 20^o, 25^o), and motion blur (5%, 10%, 15%, 20%, 25%).

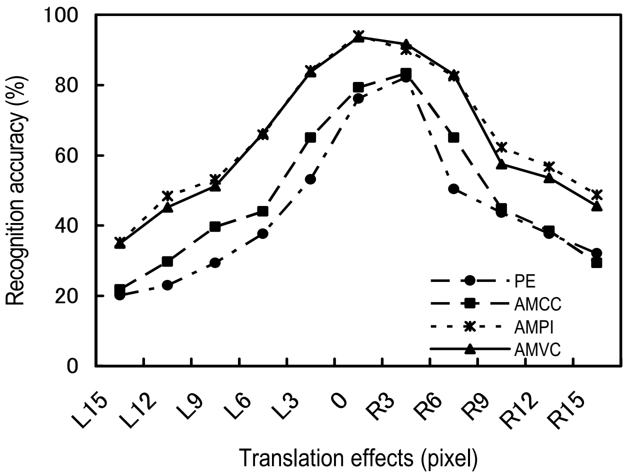

Figure 4. Recognition accuracies of images with left (L) and right (R) translation effects.

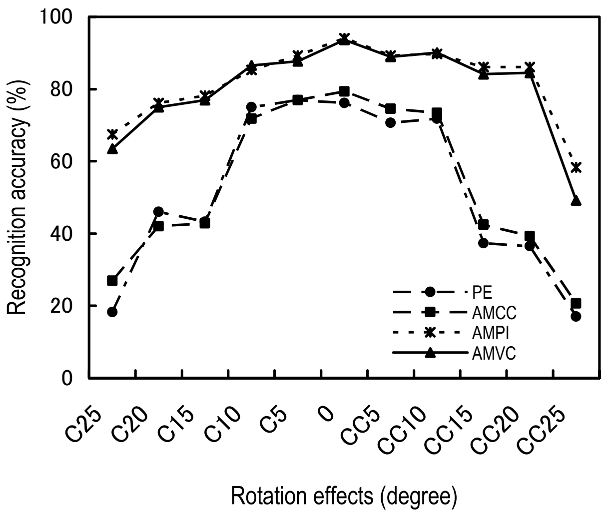

Figure 5. Recognition accuracies of images with clockwise (C) and counter-clockwise (CC) rotation effects.

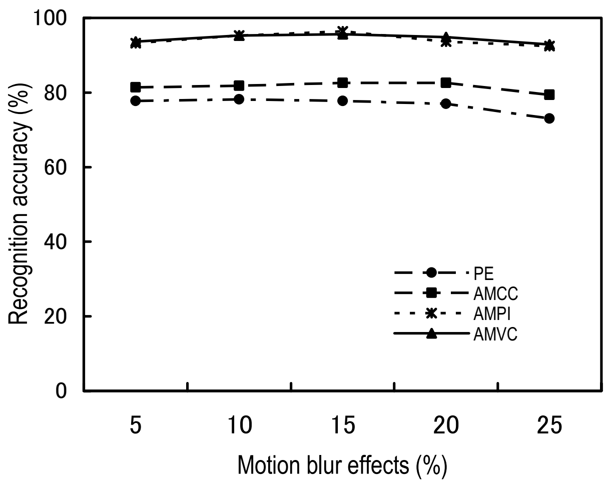

Figure 6. Recognition accuracies of images with motion blur effects.

Next, those images were projected into the eigenspace, and the appearance manifolds were created based on each construction method, as explained in section 2.2. We used spline interpolation technique to interpolate the mean vectors and linear interpolation technique to interpolate the covariance matrices.

Finally, we tested the system with input images that were different from the learning images (5^o, 15^o, 25^o…355^o) in horizontal poses and influenced by various types of degradation effects. For classification, we applied the Mahalanobis distance, as explained in section 2.3.

Figures 4, 5, and 6 show a series of the recognition accuracies of four methods in recognizing images influenced with translation effects, rotation effects, and motion blur effects, respectively. All figures indicate that the AMPI method and AMVC method, with their view-dependent covariance matrices, always achieved higher recognition accuracies than the PE method or AMCC method. For recognizing non-degraded images, the AMPI method achieved 94.05%, while the AMVC method achieved 93.65% recognition accuracy. When recognizing images with 3 pixels of right translation effects, the AMPI method achieved 90.08%, while the AMVC method achieved 91.67%. Furthermore, the AMPI method achieved 89.68% while the AMVC method achieved 90.08% when recognizing images with 10-degree counter-clockwise rotation effects.

4. Conclusion and Future Works

In this paper, we presented the use of an appearance manifold with an embedded covariance matrix as a technique to recognize 3D objects from images that are influenced by geometric and quality-degraded effects. Our proposed appearance manifold with view-dependent covariance matrix method could outperform the accuracy of the simple appearance manifold method. Moreover, performing direct covariance matrix interpolation for approximation in the AMVC method by some means worked effectively and efficiently for a relatively small interval of learning pose.

Our future works include recognizing 3D objects from images that are influenced by other types of degradation effects, as well as developing a recognition system that uses fewer learning image samples by implementing a larger interval of viewpoint orientations.

References

[1] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human face", IEEE Trans. PAMI, vol. 12, no. 1, pp. 103-108, 1990.

[2] M. Turk and A. Pentland, "Face recognition using Eigenfaces", Proc. IEEE CCVP, pp. 586-591, 1991.

[3] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation", IEEE Trans. PAMI, vol. 19, no. 7, pp. 696-710, 1997.

[4] H. Murase and S. K. Nayar. Illumination planning for object recognition using parametric eigenspaces, IEEE Trans. PAMI, vol. 16, no. 12, pp. 1219-1227, 1994.