The secret of rotating object images
— Using cyclic permutation for view-based pose estimation —

Toru Tamaki
Hiroshima University, Japan
tamaki@ieee.org

Toshiyuki Amano
NAIST, Japan
amano@is.naist.jp

Kazufumi Kaneda
Hiroshima University, Japan
kin@hiroshima-u.ac.jp

Abstract

In this paper, we propose a novel pose estimation method for a cyclic image squence of a rotating object with subspace by block diagonalization of a matrix representing transformation from an image to another. The transformation by the matrix is formulated as the action of cyclic group, and the power of a block diagonal matrix represents pose and appearance change in the sequence. Distance-based and angle-based methods are proposed to estimate pose. Experimental results with real image sequences of COIL-20 demonstrate that the subspace proposed in this paper is useful for pose estimation.

1 Introduction

When a three dimensional object rotates about an axis (as shown in Fig.1), the sequence of images of the object is cyclic: the last image is followed by the first image. When we have such a sequence of n images $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}$, the cyclic property is represented by cyclic group:

$$\boldsymbol{x}_{j+1 \bmod n} = G\boldsymbol{x}_j.$$

G is an element of a cyclic group, however, we can think it is a matrix. This relationship is essential for images of one parameter rotation, but no attentions have been paid. We propose to use the cyclic property for view-based pose estimation by linear subspace approach.

1.1 Related works

Estimation of pose of an object in an image is an important task in computer vision and pattern recognition, and methods are categorized into model-based and view-based. Model-based methods, such as [8], assume a model is given: such as known object shape, rigid motion, and projections. This approach estimates 3DOF (degrees of freedom) rotation of objects, however, requires precise geometry and restricted scene where the model can be applied. On the other hand, the advantage of view-based or appearance-based methods is to use just images of the object and make no assumptions about shape of objects and projections from 3-D to 2-D. Although it is difficult to deal with 3DOF rotation, many studies have been done even if the rotation is 1DOF (one parameter rotation).

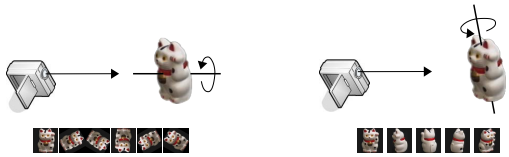

Figure 1. Images of an object by (a) in-plane and (b) off-the-plane rotation.

A major view-based method is Parametric Eigenspace method proposed by Murase et al.[10]. It learns Eigenspace of images of an object with continuously changing pose parameters. This method has been applied in a variety of areas and demonstrated its usefulness. However, there are practical problems including that it is not easy to extend the expression of spline to many (more than 2) DOF, and the search over a spline curve/surface is not closed-form but an iterative search involving expensive computation. And a theoretical question arises: *what is the Eigenspace or subspace of images of a 3D object rotating about an axis?*

For some special cases, analyses has been developed. Uenohara et al. [19, 5] proposed an efficient computation of Eigenspace for images rotating about the optical axis (so just two dimensional image rotation, or *in-plane rotation* as shown in Fig.1(a)) by using DCT or DFT[13]. Chang et al.[3] showed the same result for translational shift. Jorgan et al.[7, 5] extended for images of multiple objects rotating in-plane. Sengel et al.[17] considered in the limit when the number of images is infinite for Jorgan's method[7], then estimated a pose parameter directly with $\arctan$.

It is not easy for *off-the-plane rotation*: when an object is rotated about an arbitrary axis in three dimensional space (see Fig.1(b)). In the case of in-plane rotation, appearance of an object in images basically does not change. But in three dimensional rotation, even for 1DOF, it is impossible to find eigenvectors analytically because the appearance change depends on many properties of an object, such as shape, reflectance, shadow and etc. Therefore, many researches have been done with kernel methods or nonlinear manifold learning such as [21, 18]. Gabriele [14, 15] proposed feature-based pose estimation and view generation with elaborated grid graph representation with Gabor jets. Zhao et al.[21] used kernel PCA instead of linear PCA[10], and recently Vik et al.[20] proposed non-Gaussian modeling of appearance subspace with a method similar with [10].

However, there are few linear subspace approaches while it is still important[4]. Chang et al.[3] demonstrated to compute eigenvectors for synthetic images of an 3D cylinder just painted in black and white and rotated about an axis, then observed that eigenvectors of the images are similar with cosines. Sengel et al.[17] handled appearance changes in images of a rotating object as different image templates, but continuous pose parameters are not estimated.

A subspace approach for off-the-plane rotation was proposed by Okatani et al.[12]. They applied linear regression to the problem: first relates images with parameters by a linear map (matrix), and estimates the matrix by using pseudoinverse, then parameters are estimated by applying the matrix to an image of novel view. Amano et al.[1] used a variation of pseudoinverse with dimensionality reduction of Eigenspace of images, then estimated pose parameter linearly. Some authors use kernel methods: Ando et al.[2] used support vector regression instead of linear regression for 3DOF rotation, and Melzer et al.[9] employed kernel canonical correlation analysis (kernel CCA) for 2DOF.

These regression-like methods have shown their ability of pose estimation. However, they do not explain how the images are represented in a subspace. The answer has been shown for in-plane rotation by analytically obtained eigenvectors, but still not for off-the-plane rotation.

1.2 Our approach

In this paper, we propose a novel approach for off-the-plane rotation with a cyclic group acting on an image sequence. As mentioned above, analytical methods derived eigenvectors of images of in-plane rotating object, while regression methods used a matrix between images and parameters for off-the-plane rotation. In contrast, the proposed method focuses on the transformation from an image to another in an image sequence of off-the-plane 1DOF rotation in three dimensional space. The transformation can be seen as cyclic group, and we represent it as a matrix decomposed by block diagonalization. The main contribution of this paper is to show that the appearance change in an off-the-plane sequence can be realized by the power of the block diagonal matrix discussed from the view point of subspace. This have never been done by regression/CCA subspace methods or analytical Eigenspace methods.

2 Formulation of appearance change in image sequence with cyclic permutation

2.1 Matrix representation of relationship between images

We represent a relationship of n images in a given image sequence $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}$. The images are taken by rotating an object about an axis in three dimensional space (i.e., off-the-plane rotation), and each image $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \ldots, x_{jN})^T \in \mathbb{R}^N$ is a N dimensional vector taken at angle[1] $\theta_j = 2j\pi/n$. Throughout the paper, we assume $N > n$, the number of pixels in the images is larger than the number of images.

First we consider the following matrix G that transforms an image vector $\boldsymbol{x}_j$ into $\boldsymbol{x}_{j+1}$:

$$\boldsymbol{x}_{j+1 \bmod n} = G\boldsymbol{x}_j, \quad \boldsymbol{x}_j = G^j\boldsymbol{x}_0, \quad \boldsymbol{x}_j = G^n\boldsymbol{x}_j. \quad (1)$$

This transformation is the result of the action by a cyclic group $G_n = \{G, G^2, \ldots, G^n\}$ of degree n acting from left side on the image sequence. G is called a generator (or primitive element) of G_n, and G^n is an identity element. The group theory is an abstract concept, however, we focus only on linear transformation: that is, throughout the paper, $G \in \mathbb{R}^{N\times N}$ is a matrix and $\boldsymbol{x} \in \mathbb{R}^N$ is a vector.

However, one can ask the question: *Why can you obtain the jth image $\boldsymbol{x}_j$ from the first image $\boldsymbol{x}_0$ by just multiplying a matrix j times? When $\boldsymbol{x}_0$ is the frontal pose and $\boldsymbol{x}_j$ is the back, due to occlusions and so, does not $\boldsymbol{x}_j$ have any common information with $\boldsymbol{x}_0$?* The answer is below.

The transform can be written in a matrix form as follows:

$$[\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_{n-1}\ \boldsymbol{x}_0] = G[\boldsymbol{x}_0\ \boldsymbol{x}_1\ \cdots\ \boldsymbol{x}_{n-2}\ \boldsymbol{x}_{n-1}], \quad (2)$$

or

$$X_1 = GX_0, \quad (3)$$

where

$$X_1 = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_{n-1}\ \boldsymbol{x}_0], \quad (4)$$

$$X_0 = [\boldsymbol{x}_0\ \boldsymbol{x}_1\ \cdots\ \boldsymbol{x}_{n-2}\ \boldsymbol{x}_{n-1}]. \quad (5)$$

[1] For simplicity, the angles are evenly spaced. If the angles are irregularly sampled, the linear function $\theta(j) = 2j\pi/n$ is replaced with an appropriate nonlinear function such as piecewise linear functions or a spline curve.

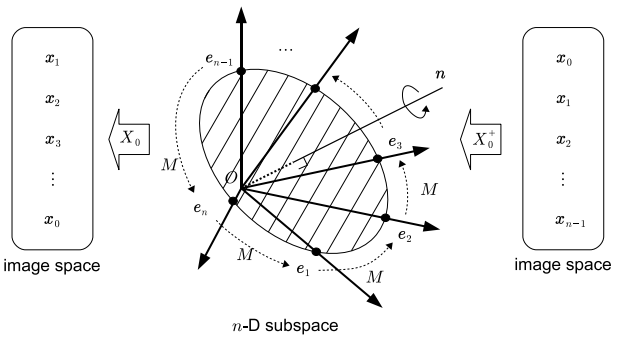

Figure 2. Two projections X_0^+, X_0 and a rotation M composing the transformation G.

Here we obtain G with X_0^+, a Moore-Penrose generalized (pseudo) inverse of X_0 with the singular value decomposition (SVD) $X_0 = E\Sigma V^T$, as follows:

$$G = X_1 X_0^+, \quad X_0^+ = (X_0^T X_0)^{-1} X_0^T = V\Sigma^{-1} E^T. \quad (6)$$

Therefore, the answer of the question above is that the matrix G indeed transforms $\boldsymbol{x}_0$ to $\boldsymbol{x}_j$ whatever the geometry of an object in the images is[2]. The reason is that Eq.(3) is an under-determined system because $N > n$. Of course the pseudoinverse in Eq.(6) is not a unique[3] and many pseudoinverses hold Eq.(3), however, this is not a problem but a necessary condition that Eq.(1) and Eq.(3) exactly hold.

When we consider the transformation from X_0 to X_1, it can be represented with a $n \times n$ column permutation matrix M multiplied from right side of X_0:

$$X_1 = X_0 \begin{pmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ & 1 & 0 & & \\ & & \ddots & \ddots & \\ & & & 1 & 0 \\ & & & & 1 \; 0 \end{pmatrix} = X_0 M, \quad (7)$$

then Eq.(6) is rewritten as follows:

$$G = X_0 M X_0^+. \quad (8)$$

2.2 Two projections and a rotation

With Eq.(8), it is interesting that we can interpret G as an combination of projections to an subspace and a rotation in the subspace. See Fig.2.

First G transforms the sequence X_0 into I_n ($n \times n$ identity matrix) because of $X_0^+ X_0 = I_n$. This means $\boldsymbol{x}_{j-1} \mapsto \boldsymbol{e}_j$, i.e., each image $\boldsymbol{x}_{j-1}$ is mapped to a canonical unit vector $\boldsymbol{e}_j$ in which all components are 0 except j th component is 1. Next, M moves the unit vector $\boldsymbol{e}_j$ to $\boldsymbol{e}_{j+1}$. This can be done by just shifting components in $\boldsymbol{e}_j$, but M is indeed rotation about the axis $\boldsymbol{n} = (1, 1, \ldots, 1) \in \mathbb{R}^n$ and makes the unit vector form a locus of a hypercircle on a hyperplane[4] in $\mathbb{R}^n$. Finally X_0 projects vectors back to the image space from the subspace.

Therefore, the images in the sequence are projected onto the circle in the subspace, and well separated with distance $\sqrt{2}$ from each other[5], and transfered from one to the next by M.

For recognizing unknown pose between learned poses, the concept of the proposed method is to extend this *discrete rotation* M into *continuous rotation* by interpolating M with block diagonalization discussed below.

[2]Imagine how large G is — $N \times N$! Even when G is decomposed, U_1 and U_2 are the same size with X_0. Therefore, G has so enough elements that represent information between images even if $\boldsymbol{x}_0$ and $\boldsymbol{x}_j$ do not have.

[3]It is unique in the sense that a minimum norm solution is given.

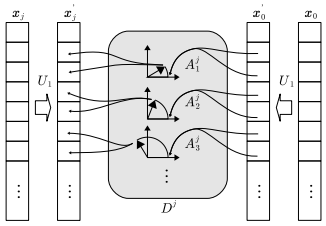

Figure 3. Rotations by A_k^j in 2-D subspaces.

2.3 Decomposition of G

M is decomposed with a real block diagonal matrix D and a real orthogonal matrix W as $M = WDW^T$. Then, the decomposition of G is

$$G = U_2 D U_1, \quad U_1 = W^T X_0^+, \quad U_2 = X_0 W, \quad (9)$$

where

$$D = \begin{pmatrix} 1 & & & \\ & A_1 & & \\ & & A_2 & \\ & & & \ddots \end{pmatrix}, \quad A_k \in \mathbb{R}^{2\times 2}, \quad (10)$$

D has 2×2 blocks A_k at its diagonal part. See appendix for the detail of the block diagonalization.

With $U_1 U_2 = I_n$, the transformation from $\boldsymbol{x}_0$ to $\boldsymbol{x}_j$ can be represented as

$$\boldsymbol{x}_j = U_2 D^j U_1 \boldsymbol{x}_0, \quad (11)$$

instead of $\boldsymbol{x}_j = G^j \boldsymbol{x}_0$.

Here, the matrix U_1 can be regarded as a projection from the image space onto a n-dimensional subspace representing the pose of an object in the images. See Fig.3 in which we call $\boldsymbol{x}' = U_1 \boldsymbol{x}$ *an image in the subspace*. Each pair of

[4]It is perpendicular to $\boldsymbol{n}$, and the distance to the origin is $\frac{1}{\sqrt{n}}$.

[5]$\forall j, k, j \neq k \Rightarrow ||\boldsymbol{e}_j - \boldsymbol{e}_k|| = \sqrt{1+1} = \sqrt{2}$.

row vectors of U_1 corresponding a 2×2 block A_k of D is a linear projection from the image space onto two dimensional (2-D) subspace spanned by the row vectors. These 2-D subspaces are independent and orthogonal to each other because all blocks do not overlap. Therefore, the projection of an original image is a set of projections onto different 2-D subspaces, and multiplying D in the subspace means 2-D rotations (with A_k by θ_k) of 2-D vectors comprised of two pixels of the image in the subspace.

2.4 Demonstrating the subspace

As the derivation above, the matrix G transform an image to another in the image sequence X_0 by the power of G:

$$\boldsymbol{x}_j = G^j \boldsymbol{x}_0, \quad \text{or} \quad \boldsymbol{x}_j = U_2 D^j U_1 \boldsymbol{x}_0. \tag{12}$$

Therefore, j (the power of D^j) decides how much the image $\boldsymbol{x}_0$ is transformed in the image sequence.

Now we are interested in not only observing the transformation from $\boldsymbol{x}_0$ to $\boldsymbol{x}_j$ but also extending the range of the power j from several integer numbers $(0, 1, \ldots, n-1)$ to a real interval $[0, n[$.

For an off-the-plane rotation sequence, Fig.4(a) demonstrates an example using object 4 in COIL-20 [11]. 36 images including $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$ (0, 10, 20[deg]) are used for learning. Two images (5,15 [deg]) corresponding to $\boldsymbol{x}_{0.5}, \boldsymbol{x}_{1.5}$ are shown for comparison. The lower row shows images $\boldsymbol{x}_{0.1j}$ created by

$$\boldsymbol{x}_{0.1j} = G^{0.1j} \boldsymbol{x}_0 = U_2 D^{0.1j} U_1 \boldsymbol{x}_0, \tag{13}$$

for $j = 0, 1, 2, \ldots, 20$. The created images $\boldsymbol{x}_{1.0}$ and $\boldsymbol{x}_{2.0}$ are exactly same with the learned images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. For the other images between learned images, especially $\boldsymbol{x}_{0.5}$ and $\boldsymbol{x}_{1.5}$ for comparison, the appearance are very similar with actual intermediate images. Actually they look like one made by blending two learned images, but our objective is not to make created images close to the real ones, but to utilize them for pose estimation as shown in the next section.

Although the proposed method is formulated for a single axis rotation, Eq.(3) can be applicable to any revolving image sequence such as a light turns around in front of a face. Fig.4(b) illustrates such an example for different light directions. 20 face images of P00 in the Yale Face Database B [6] including $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4, \boldsymbol{x}_5$ (cropped) are used for learning. The lower row shows images are created by $\boldsymbol{x}_{0.25j} = G^{0.25j} \boldsymbol{x}_0$. Discussion on the estimation of light direction is out of scope of this paper, but this example implies that the proposed method can be used for estimating illumination change.

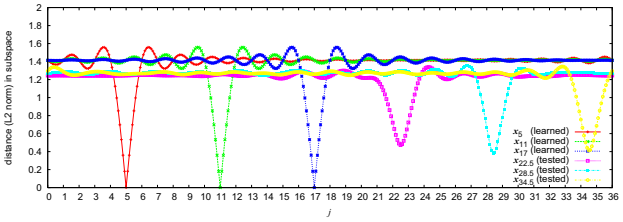

Figure 5. Euclidean distance in the subspace between $D^j \boldsymbol{x}'_0$ and $\boldsymbol{x}'_j$ (learned $j = 5, 11, 17$, and not learned $j = 22.5, 28.5, 34.5$). Horizontal axis is the power j of D^j.

3 Estimation of pose of an object in novel view

In this section, we propose two methods for estimation of pose of a new image by using the subspace described in the previous section.

3.1 Estimation by distance in the subspace D

As shown at the end of the last section, we have shown that extending real numbers of the power j of D^j gives images between learned samples.

Here we make an assumption that *a novel image $\boldsymbol{x}$ is matched with $G^j \boldsymbol{x}_0$ for some j and this also holds for images in the subspace: $\boldsymbol{x}'$ is matched for some j with $D^j \boldsymbol{x}'_0$ in the subspace*, where $\boldsymbol{x}' = U_1 \boldsymbol{x}$ and $'$ denotes an image in the subspace. For matching, we use the Euclidean distance in the subspace:

$$j = \underset{j \in [0,n[}{\operatorname{argmin}} ||\boldsymbol{x}' - D^j \boldsymbol{x}'_0||^2, \tag{14}$$

$$\theta = j\theta_1 = \frac{2\pi}{n} j. \tag{15}$$

See appendix for θ_1 and constructing D^j.

The estimation performs exhaustive search for j and it seems to be computationally expensive. However, we can use an effective algorithm for the search by using coarse-to-fine strategy. Fig.5 shows distances in the subspace by Eq.(15) for some real image sequence (see the later section for details). We can observe that the distances have sharp minima at corresponding j for learned images. Even for images not used for the learning, the distances have smooth minima around correct j. Based on this observation, first we search a minimum of j with a large step, then find around the minimum again with more smaller step, and gradually the interval of search shrinks. This strategy decreases computational cost and achieves estimation at any precision.

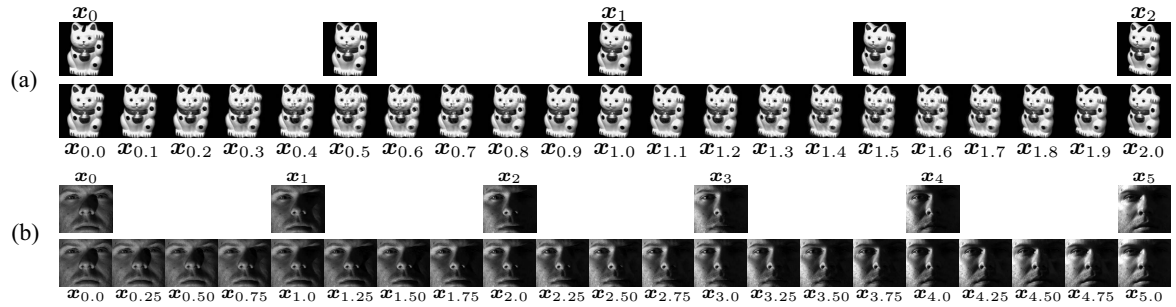

Figure 4. Images created by repeatedly multiplying a matrix G^j to the first image x_0. (a) images of off-the-plane rotation from COIL-20. $G^j = G^{0.1}$. (b) images of changing light direction from Yale Face Database B. $G^j = G^{0.25}$. Upper row shows learned images, and lower row shows created images between each learned images. *Note that supplemental full-length movies are attached/embedded in this PDF file (use Adobe Reader to see it).*

3.2 Estimation by angle of vectors in a 2D subspace A_1

The estimation method described above involves iterative search for minimum even if there is the efficient algorithm. Here we propose a direct estimation method without any searching. As mentioned before, an image is projected by U_1 onto many different 2-D subspaces in which a 2-D vector of two pixels is rotated by A_k. Now we focus on two pixels corresponding A_1 where the pair of pixels in two learned images next to each other, $\boldsymbol{x}_j$ and $\boldsymbol{x}_{j+1}$, have the angle θ_1, the incremental step of the rotation.

So we propose to estimate a pose parameter for a novel image in the subspace $\boldsymbol{x}'$ with $\boldsymbol{x}'_0$ by using the angle subtended by two 2-D vectors, $\boldsymbol{x}'', \boldsymbol{x}''_0 \in \mathbb{R}^2$, corresponding to the 2-D subspace of A_1. To extract a 2-D vector $\boldsymbol{x}'' \in \mathbb{R}^2$ from $\boldsymbol{x}' \in \mathbb{R}^n$ corresponding to A_1, just multiply the following $2 \times n$ matrix:

$$\boldsymbol{x}'' = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} \boldsymbol{x}'. \tag{16}$$

But $\boldsymbol{x}' = U_1 \boldsymbol{x}$ is substituted above, the 2-D vector $\boldsymbol{x}'' \in \mathbb{R}^2$ is directly extracted from $\boldsymbol{x} \in \mathbb{R}^N$ by combining U_1 and the $2 \times n$ matrix:

$$\boldsymbol{x}'' = U'_1 \boldsymbol{x}, \tag{17}$$

$$U'_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix} U_1, \tag{18}$$

and $\boldsymbol{x}''_0 \in \mathbb{R}^2$ is extracted:

$$\boldsymbol{x}''_0 = A_1 U'_1 \boldsymbol{x}_0. \tag{19}$$

Here U'_1 is the 2-D subspace proposed in this paper for estimating the pose angle.

Figure 6. A part of images used for the experiments. $x_0, x_1, \ldots$ are learned (with box marks), $x_{0.5}, x_{1.5}, \ldots$ are tested images.

The angle θ subtented by the two 2-D vectors is calculated with $\cos\theta$ and $\sin\theta$. The innter product between $\boldsymbol{x}''$ and $\boldsymbol{x}''_0$ computes $\cos\theta$:

$$\cos\theta = \frac{\boldsymbol{x}''^T_0 \boldsymbol{x}''}{||\boldsymbol{x}''_0||\,||\boldsymbol{x}''||}. \tag{20}$$

$\sin\theta$ is computed by cross product with two 3-D vectors extented with 0:

$$\boldsymbol{x}'''_0 = (\boldsymbol{x}''^T_0, 0)^T \in \mathbb{R}^3, \tag{21}$$

$$\boldsymbol{x}''' = (\boldsymbol{x}''^T, 0)^T \in \mathbb{R}^3, \tag{22}$$

$$(0, 0, \sin\theta)^T = \frac{\boldsymbol{x}''' \times \boldsymbol{x}'''_0}{||\boldsymbol{x}''_0||\,||\boldsymbol{x}''||}. \tag{23}$$

Then, $\theta = \tan^{-1}\left(\frac{\sin\theta}{\cos\theta}\right)$ is the angle between $\boldsymbol{x}''$ and $\boldsymbol{x}''_0$, then the estimate of the pose of the image $\boldsymbol{x}$.

4 Experimental results

We implemented the proposed method with Scilab-4.1 and evaluated with a real image sequence of the object 4 (the cat) from COIL-20[11]. The 72 images are $N = 128 \times 128$ in size, taken by rotating the object by 5 degrees each (see Fig.6). The rotation of the images is 1DOF (single axis rotation), but it is off-the-plane rotation because the axis is not the optical axis of the camera.

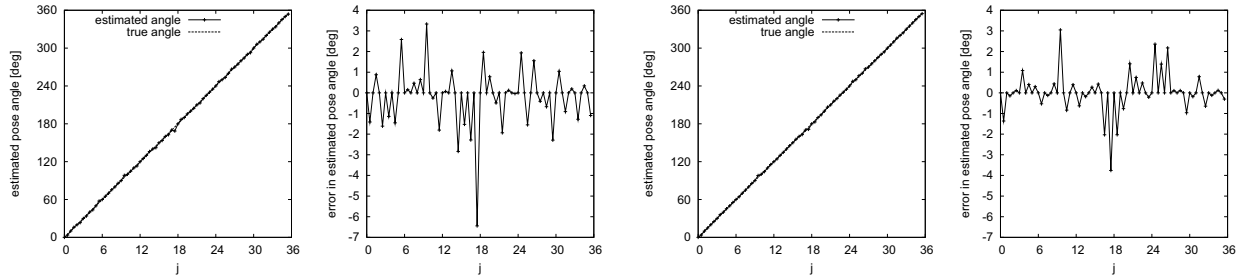

Figure 7. Estimation results with (a)(b) distance-based and (c)(d) angle-based method for images $\boldsymbol{x}_j$ $(j = 0, 0.5, 1, 1.5, 2, \ldots, 35.5)$. (a)(c) Estimated pose.

Table 1. RMSEs with two methods for 20 objects in COIL-20 (in [deg]).

object No.	1	2	3	4	5	6	7	8	9	10	
with distance	1.21	1.39	1.56	1.80	1.23	29.71	1.44	1.60	1.05	1.53	
with angle	0.73	1.69	3.84	1.23	2.55	7.33	2.97	2.69	6.74	1.66	
object No.	11	12	13	14	15	16	17	18	19	20	average
with distance	1.58	23.05	1.03	1.90	8.33	8.83	7.71	13.10	1.64	3.32	5.65
with angle	5.78	4.18	3.11	7.38	2.01	1.53	2.35	3.64	6.99	2.21	3.53

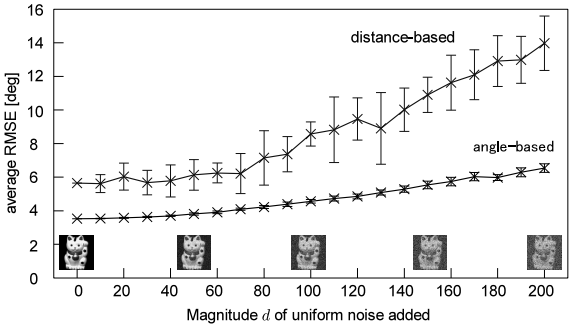

Figure 8. RMSEs of estimation with std. for 10 trials for noisy images. Horizontal axis is the magnitude $[-d, d]$ of uniform noise added. Vertical axis is average RMSE for 20 objects for only images not learned, but with noise.

For learning eigenspace and computing U_1, we used 36 images corresponding to 0, 10, 20, ... degrees as images $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{35}$ in the experiment. Therefore, $\theta_1 = 10$[deg] in this experiment. Another 36 images corresponding to $5, 15, 25, \ldots$ degrees were used not for learning but for evaluation as images $\boldsymbol{x}_{0.5}, \boldsymbol{x}_{1.5}, \ldots, \boldsymbol{x}_{35.5}$.

To illustrate properties of the subspace, we computed Euclidean distances between learned images $\boldsymbol{x}'$ and the image $\boldsymbol{x}'_0$ rotated by the power of D in the subspace. Fig.5 shows the distances, and the horizontal axis is the power j of D^j, and the vertical axis is the Euclidean distance. For example, the distance with $\boldsymbol{x}_5$ is $||\boldsymbol{x}'_5 - D^j\boldsymbol{x}'_0||^2$ and has a sharp minimum at $j = 5$ which means that the subspace is well learned. The distances with $\boldsymbol{x}'_5$ and the other learned images $\boldsymbol{x}'_j$, or equivalently $D^j\boldsymbol{x}'_0$, are all the same distance, $\sqrt{2}$. When the power j is a real number, the distance deviates from $\sqrt{2}$ and seems to be an interpolated curve comprised of sinusoids with different frequencies. The deviation from $\sqrt{2}$ (or ripple width) is so small that the search for minimum is not affected.

Fig.5 also shows distances with images not used for learning. Even if the images are not learned, the distance have smooth minimum around correct power. This means that the distance in the subspace is useful for the pose estimation.

Next, in Fig.7(a)(b) we show result of pose estimation with the method described in section 3.1, the search for minimum of j with the distance. Correct poses for the learned images $\boldsymbol{x}_j$ $(j = 0, 1, 2, \ldots)$ are estimated with no error. Poses for the images not learned $\boldsymbol{x}_j$ $(j = 0.5, 1.5, 2.5, \ldots)$ are also estimated well. The maximum error is about 7[deg], and almost less than $\pm$2[deg], and RMSE (root mean squared error) for tested images only (not including learned images) is 1.80[deg]. Fig.7(c)(d) shows estimation result with the method described in section 3.2, the use of angle of two vectors in 2-D subspace. The maximum error is about 4[deg], and RMSE is 1.23[deg]. This means that the angle-based method is better than the distance-based method, and the angle of the two vectors in the 2-D subspace well represents the pose of the object in an image.

This is supported by estimation results shown in Tab.1 for all 20 objects in COIL-20 with both distance-based and angle-based methods. The result of Fig.7 is shown at object No. 4 in Tab.1. In average, RMSE of the angle-based method (3.53[deg]) is smaller than that of the distance-based (5.65[deg]).

Fig.8 shows the robustness of the angle-based method for noisy images shown. These images are contaminated by uniform noise up to ± 200 without any intensity normalization (negative pixel values and large values are just used) where the range of pixel value in original images is between 0 and 255. Even when ± 200 uniform noise is added, the average RMSE of angle-based method is less than 7[deg], while error of the distance-based method increases larger than 14 [deg]. This result demonstrates how robust the angle-based method is as well as the subspace proposed in this paper is useful for pose estimation. Note that for clutterd images (e.g., objects are occluded by a black rectangle), the proposed angle-based method has shown a good performance (not shown in this paper).

5 Conclusions

We have proposed a novel framework with cyclic group for appearance change in an image sequence of a rotating (1DOF but off-the-plane) object in 3-D. The proposed method constructs a subspace by block diagonalization of a matrix that represents cyclic group acting on the image seqeunce and transforms an image to another in the sequence. We have shown how the power of the block diagonal matrix produces the transform between images in and not in the sequence, then proposed two methods to estimate pose of a novel image; distance-based and angle-based. Experimental results with real image sequences demonstrated that the angle-based method is robust against noise and better than the distance-based method. The experiments are still limited, and comparisons with conventional methods are planned for the future.

Some limitations of the proposed method should be noticed. First, the method is applicable to sequences in which an object in images are revolutionary rotated: for example, a face sequence taken from left side to right side with frontal face has no images of the back of the head, so it is not applicable. Second, it seems to be difficult to extend the proposed method to handle with 3DOF rotation of an object. These are caused by the use of the matrix G as a cyclic group. Therefore, future works include to find an appropriate representation of relationship between such image sequences with group theory for extending application area of the proposed method. And also we have to investigate the pseudoinverse used in the derivation that is theoretically not determined uniquely because we assume $N > n$. It is clear that the linear map defined by the pseudoinverse is crucial to improve generalization and decrease estimation error for unknown pose.

Acknowledgments We thank Hiroshi Tamaru and Yumiko Ichihara at Hiroshima University for discussing the distance between images in the subspace. Also we would like to thank Hitoshi Sakano at NTT Data for his comments to improve the quality of the paper.

A Complex diagonalization of M

A $n \times n$ permutation matrix M to be diagonalized and its characteristic equation are[16, 3, 13, 7]:

$$M = \begin{pmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ & 1 & 0 & & \\ & & \ddots & \ddots & \\ & & & 1 & 0 \\ & & & & 1 \end{pmatrix},$$

$$|M - \lambda I| = \begin{vmatrix} \lambda & & & & -1 \\ -1 & \lambda & & & \\ & -1 & \lambda & & \\ & & \ddots & \ddots & \\ & & & -1 & \lambda \\ & & & & -1 \end{vmatrix} = \lambda^n - 1,$$

so the eigenvalues λ are n different primitive n-th roots of unity ζ_n:

$$\lambda_k = \sqrt[n]{1} = \zeta_n^k = e^{\frac{2k\pi}{n} i}, \quad k = 0, 1, 2, \ldots, n-1,$$

where $i = \sqrt{-1}$. Let $\boldsymbol{w}_k = (w_1, w_2, \ldots, w_n)^T$ be the eigenvector corresponding ζ_n^k, then

$$M\boldsymbol{w}_k = \zeta_n^k \boldsymbol{w}_k$$
$$(w_n, w_1, w_2, \ldots, w_{n-1})^T = (\zeta_n^k w_1, \zeta_n^k w_2, \ldots, \zeta_n^k w_n)^T.$$

Therefore, the eigenvector is

$$\boldsymbol{w}_k = (\zeta_n^{(n-1)k}, \ldots, \zeta_n^{2k}, \zeta_n^k, 1)^T,$$

and M is diagonalized as $M = W' D' W'^H$ with:

$$\begin{aligned} D' &= \mathrm{diag}(1, \zeta_n, \zeta_n^2, \ldots, \zeta_n^{n-1}), \\ W' &= (\boldsymbol{w}_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{n-1}), \end{aligned} \tag{24}$$

where ^{H} denotes complex conjugate and W' is the basis of complex DFT (Discrete Fourier Transform) [13].

B Real block diagonalization of M

Next, block diagonalization of M is shown[16, 3, 13]. ζ_n^k and ζ_n^{n-k}, eigenvalues of M, are complex conjugate to each other. To make corresponding complex conjugate eigenvectors $\boldsymbol{w}_k, \boldsymbol{w}_{n-k}$ real vectors, dividing them into real and imaginary parts:

$$\boldsymbol{w}_k = \frac{1}{\sqrt{2}}(\boldsymbol{c}_k + i\boldsymbol{s}_k), \quad \boldsymbol{w}_{n-k} = \frac{1}{\sqrt{2}}(\boldsymbol{c}_k - i\boldsymbol{s}_k).$$

Then, the multiplication of M with the vectors

$$M(\boldsymbol{w}_k, \boldsymbol{w}_{n-k}) = (\boldsymbol{w}_k, \boldsymbol{w}_{n-k}) \begin{pmatrix} \zeta_n^k & 0 \\ 0 & \zeta_n^{n-k} \end{pmatrix},$$

is rewritten with $\zeta_n^k = \cos\theta_k + i \sin\theta_k$ as follows:

$$\begin{aligned} M(\boldsymbol{c}_k, \boldsymbol{s}_k) &= (\boldsymbol{c}_k, \boldsymbol{s}_k) \begin{pmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{pmatrix} \\ &= (\boldsymbol{c}_k, \boldsymbol{s}_k) A_k. \end{aligned}$$

Now M is diagonalized with block diagonal matrix D as $M = WDW^T$, where

$$D = \begin{cases} \mathrm{diag}(1, A_1, A_2, \ldots, A_s), & n \text{ is odd}, \\ \mathrm{diag}(1, A_1, A_2, \ldots, A_s, -1), & n \text{ is even}, \end{cases}$$

$$W = \begin{cases} (\boldsymbol{w}_0, \boldsymbol{c}_1, \boldsymbol{s}_1, \boldsymbol{c}_2, \boldsymbol{s}_2, \ldots, \boldsymbol{c}_s, \boldsymbol{s}_s), & n \text{ is odd}, \\ (\boldsymbol{w}_0, \boldsymbol{c}_1, \boldsymbol{s}_1, \boldsymbol{c}_2, \boldsymbol{s}_2, \ldots, \boldsymbol{c}_s, \boldsymbol{s}_s, \boldsymbol{w}_{n/2}), & n \text{ is even}, \end{cases}$$

$$s = \begin{cases} \frac{n-1}{2}, & n \text{ is odd}, \\ \frac{n-2}{2}, & n \text{ is even}, \end{cases}$$

where W is the basis of DFT[13]. Note that W' and W are normalized so that norm of each column vector is 1.

C The power of D

If we need D^j, the angle in the 2×2 blocks A_k are multiplied:

$$D^j = \begin{cases} \mathrm{diag}(1, A_1^j, A_2^j, \ldots, A_s^j), & n \text{ is odd}, \\ \mathrm{diag}(1, A_1^j, A_2^j, \ldots, A_s^j, (-1)^j), & n \text{ is even}, \end{cases}$$

$$A_k^j = \begin{pmatrix} \cos j\theta_k & \sin j\theta_k \\ -\sin j\theta_k & \cos j\theta_k \end{pmatrix}.$$

Note that D^j becomes a complex matrix when n is even.

This property is the most usefull one for the proposed formulation because G^j can be calculated by just multiplying the angle θ_k with j. If you use the Jordan (normal or canonical) form as block diagonalization of M, D^j is not easy to compute. And actually all eigenvectors of M are different to each other, the Jordan form of M is equivalent to the eigendecomposition Eq.(24); no Jordan form exists for M.

References

[1] T. Amano and T. Tamaki. A fast linear pose estimation method of 3D object using EbC image pair. *IEICE Trans.*, J90-D(8):2060–2069, 2007. (in Japanese).

[2] S. Ando, Y. Kusachi, A. Suzuki, and K. Arakawa. Appearance based pose estimation of 3D object using support vector regression. *ICIP2005*, 1:I–341–344, 2005. online.

[3] C.-Y. Chang, A. Maciejewski, and V. Balakrishnan. Fast eigenspace decomposition of correlated images. *IEEE Trans. IP*, 9(9):1937–1949, 2000. online.

[4] P. Chen and D. Suter. An analysis of linear subspace approaches for computer vision and pattern recognition. *Intl. J. of Computer Vision*, 68(1):83–106, 2006. online.

[5] F. De la Torre. Component analysis for computer vision. *ECCV2006 Tutorial*, 2006. online.

[6] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. PAMI*, 23(6):643–660, 2001. online.

[7] M. Jorgan, E. Žagar, and A. Leonardis. Karhunen-Loéve expansion of a set of rotated templates. *IEEE Trans. IP*, 12(7):817–825, 2003. online.

[8] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. PAMI*, 13(5):441–450, 1991. online.

[9] T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 36:1961–1971, 2003. online.

[10] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Intl. J. of Computer Vision*, 14(1):5–24, 1995. online.

[11] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20). Technical Report CUCS-005-96, Columbia University, 1996. online.

[12] T. Okatani and K. Deguchi. Yet another appearance-based method for pose estimation based on a linear model. *IAPR Workshop on Machine Vision Applications 2000*, pages 258–261, 2000.

[13] R.-H. Park. Comments on "Optimal approximation of uniformly rotated images: Relationship between Karhunen-Loève expansion and discrete cosine transform". *IEEE Trans. IP*, 11(3):332–334, 2002. online.

[14] G. Peters. Efficient pose estimation using view-based object representations. *Machine Vision and Applications*, 16(1):59–63, 2004. online.

[15] G. Peters and C. von der Malsburg. View reconstruction by linear combination of sample views. *British Machine Vision Conference*, 1:223–232, 2001. online.

[16] I. Satake. *Linear Algebra*. Marcel Dekker Inc., 1975.

[17] M. Sengel and H. Bischof. Efficient representation of in-plane rotation within a PCA framework. *Image and Vision Computing*, 23:1051–1059, 2005. online.

[18] T. Tangkuampien and D. Suter. 3D object pose inference via kernel principal component analysis with image euclidian distance (IMED). *British Machine Vision Conference*, 1:137–146, 2006. online.

[19] M. Uenohara and T. Kanade. Optimal approximation of uniformly rotated images: Relationship between Karhunen-Loeve expansion and discrete cosine transform. *IEEE Trans. IP*, 7(1):116–119, 1998. online.

[20] T. Vik, F. Heitz, and P. Charbonnier. Robust pose estimation and recognition using non-gaussian modeling of appearance subspaces. *IEEE Trans. PAMI*, 29(5):901–905, 2007. online.

[21] L.-W. Zhao, S.-W. Luo, and L.-Z. Liao. 3D object recognition and pose estimation using kernel PCA. *Proc. of Intl. Conf. Machine Learning and Cybernetics*, 5:3258–3262, 2004. online.