Comparison between Constrained Mutual Subspace Method and Orthogonal Mutual Subspace Method From the viewpoint of orthogonalization of subspaces

著者	FUKUI Kazuhiro, YAMAGUCHI Osamu
journal or publication title	Technical Report of Department of Computer Science
page range	1-10
year	2006
シリーズ	CS-TR;06-7
URL	http://hdl.handle.net/2241/106815

Comparison between Constrained Mutual Subspace Method and Orthogonal Mutual Subspace Method

– From the viewpoint of orthogonalization of subspaces –

Kazuhiro FUKUI† and Osamu YAMAGUCHI††

† System Information Engineering, Tsukuba University Tennoudai 1–1–1, Tsukuba,Japan

† Toshiba Corporate R&D Center Komukai-Toshiba-chou 1, Saiwai-ku, Kawasaki, Japan

Abstract This paper compares the performances between constrained mutual subspace method (CMSM), orthogonal mutual subspace method (OMSM), and also between their nonlinear extensions, namely kernel CMSM (KCMSM) and kernel OMSM (KOMSM). Although the princeples of the feature extraction in these methods are different, their effectiveness are commonly derived from the orthogonalization of subspace, which is widely used to measure the performance of subspace-based methods. CMSM makes the relation between class subspaces similar to orthogonal relation by projecting the class subspaces onto the generalized difference subspaces. KCMSM is also based on this projection in the nonlinear feature space. On the other hand, OMSM orthogonalizes class subspaces directly by whitening the distribution of the class subspaces. KOMSM also utilizes this orthogonalization method in the nonlinear feature space. From the experimental results, the performances of both the kernel methods (KCMSM and KOMSM) are found to be very high as compared to their linear methods (CMSM and OMSM) and their performances levels are well in the same order in spite of their different principles of orthogonalization.

Key words Constrained mutual subspace method, orthogonal mutual subspace method

1. Introduction

This paper deals with the comparison of the constrained mutual subspace method (CMSM) [5] and the orthogonal mutual subspace method (OMSM) [6], including the nonlinear extensions of these methods with kernel trick.

These methods used here are the extensions of the mutual subspace method (MSM), which is a generalization of the subspace method (SM) [1]〜[3]. SM is used to calculate the similarity as the minimum canonical angle θ_1 between an input vector and a reference subspace which represents the variation of the learning set where an $n \times n$ image pattern is represented as a vector in $n \times n$-dimensional space $\mathcal{I}$.

In the case of MSM, the canonical angle is calculated between an input subspace and reference subspace as the similarity. MSM can deal with various recognition problems of the sets of images, such as a video image, multi-view images obtained from a multiple camera system.

MSM has a high ability for absorbing the changes of pattern to achieve higher performance compared to that of SM using a single image. Since the generated subspaces are independent to each other in the case of MSM, the classification performance is still insufficient for many applications as is the case with other methods based on PCA [3]. Although each subspace represents the distribution of the training patterns well in terms of a least mean square approximation, there is no reason to assume a priori that it is the optimal subspace in terms of classification performance.

To improve the ability of classification of MSM, the constrained mutual subspace method (CMSM) [5] and the orthogonal mutual subspace method (OMSM) [6] have been proposed. The validity of these methods is due to the orthogonalization of class subspaces.

In the case od CMSM, the projection of each class subspace is onto the generalized differenece subspace, it extracts a common subspace of all the class subspaces from each subspace, so that the canonical angles between subspaces are enlarged to approach to the orthogonal relation. On the other hand, OMSM orthogonalizes directly all the class subspaces based on the framework of the orthogonalization method proposed by Fukunaga & Koontz [19], [20].

Since the distribution of each class can not be represented by a linear subspace without overlaping with another subspace, the performance of CMSM and OMSM are poor as the methods based linear subspace space, when the distribution of patterns has high nonlinear structures.

In order to overcome this problem, it is valid to represent such the distribution by a nonlinear subspace generated by kernel nonlinear PCA [8].

It is vividly stated that SM and MSM have extended to the kernel subspace method (KSM) [11], [12] and the kernel mutual subspace method (KMSM) [14], respectively, based on the frame work of the kernel PCA. Moreover CMSM has been also extended to the kernel CMSM (KCMSM) [13] to achieve higher performance compared to KSM and KMSM.

The basis algorithm of OMSM is almost the same as that of CMSM except a part of the process of calculating similarity. Therefore in this paper, we will try to extend OMSM to the kernel nonlinear method, and name the extended method the kernel nonlinear orthogonal mutual subspace method (KOMSM).

KOMSM has an interesting characteristic that it can always orthogonalize all subspaces perfectly regardless of the number of classes and the dimension of each class subspace when the Gaussian kernel function is used in kernel trick. This is because the Gaussian kernel fucntion maps a pattern onto the infinite dimensional feature space. In contrast, the oroginal linear OMSM cannot realize the perfect orthogonalization of subspaces unless the product of the number of classes and the dimension of each subspace is less than the dimension of the linear feature space.

The nonlinear kernel methods have a lack of requiring a lot of computing cost depending the number of learning patterns and classes. Especially, the computing cost of KOMSM is extremely high, because it has to calculate all the eigenvalues and the eigenvectors of a matrix with the size equal to to the number of classes and the dimension of each class. Thus, the learning for KOMSM is more difficult as the size of the matrix becomes larger with increasing the number and the dimesion of classes.

By contrast, since KCMSM requires only a part of the eigenvectors corresponding to the larger eigenvalues of all eigenvectors of the kernel matrix, it does not have such the serious problem.

The paper is organized as follows. Firstly we outline the algorithms of MSM, CMSM and OMSM in sections 2, 3 and 4, respectively. In section 5, we describe the kernel trick for nonlinear extension of CMSM and OMSM. In section 6 we describe the algorithm of KCMSM. Then, in section 7, we extend OMSM to the nonlinear kernel OMSM (KOMSM). In section 8 we compare the performance of all the methods. The section 9 is the discussion. Finally we conclude the paper.

2. Mutual subspace method

In this section, we firstly describe the concept of canonical angle between subspaces. Next we outline the algorithm of MSM based on the canonocal angles.

2.1 Canonical angles between subspaces

Assume that $\mathcal{P}$ and $\mathcal{Q}$ are an N-dimensional input subspace and an M-dimensional reference subspaces, in the f-dimensional space $\mathcal{I}$, which are generated from the pattern set of input and reference, respectively. We can obtain N canonical angles (for convenience $N \leqq M$) between $\mathcal{P}$ and $\mathcal{Q}$ [15].

Let $\mathbf{\Phi}_i$ and $\mathbf{\Psi}_i$ denote the i-th f-dimensional orthonormal basis vectors of the subspaces $\mathcal{P}$ and $\mathcal{Q}$, respectively. The value $\cos^2\theta_i$ of the i-th smallest canonical angle θ_i ($i = 1, \ldots, N$) is obtained as the i-th largest eigenvalue λ_i of the following $N \times N$ matrix $\mathbf{X}$ [5], [15]:

$$\mathbf{X}\mathbf{c} = \lambda \mathbf{c} \tag{1}$$

$$\mathbf{X} = (x_{ij}),\ x_{ij} = \textstyle\sum_{k=1}^{M} (\mathbf{\Psi}_i \cdot \mathbf{\Phi}_k)(\mathbf{\Phi}_k \cdot \mathbf{\Psi}_j)$$

The maximum eigenvalue of these matrices represents $cos^2\theta_1$ of the minimum canonical angle θ_1, whereas the second maximum eigenvalue represented $cos^2\theta_2$ of the minimum angle θ_2 in the direction perpendicular to that of θ_1. $cos^2\theta_i$ for $i = 3, \ldots, N$, are calculated similarly.

$$cos^2\theta_i = \max_{\substack{\mathbf{u}_i \perp \mathbf{u}_{j(=1,\ldots,i-1)} \\ \mathbf{v}_i \perp \mathbf{v}_{j(=1,\ldots,i-1)}}} \frac{|(\mathbf{u}_i \cdot \mathbf{v}_i)|^2}{||\mathbf{u}_i||^2 ||\mathbf{v}_i||^2}\ , \tag{2}$$

where $\mathbf{u}_i \in P, \mathbf{v}_i \in Q, ||\mathbf{u}_i|| \neq 0, ||\mathbf{v}_i|| \neq 0$, $(\cdot)$ and $||\cdot||$ represent an inner product and a norm respectively.

2.2 Definition of similarity between two subspaces

We consider the value of the mean of the canonical angles, $S[t] = \frac{1}{t}\sum_{i=1}^{t} \cos^2\theta_i$, as the similarity between two subspaces. The similarity $S[t]$ has the following characteristics. In the case that two subspaces coincide completely with each other, $S[t]$ is 1.0, since all canonical angles are zero. The similarity $S[t]$ becomes smaller as the two subspaces separate. Finally, the similarity $S[t]$ is zero when the two subspaces are orthogonal to each other. Note that the value $S[t]$ reflects the structual similarity between two subspaces.

3. Constrained mutual subspace method

To improve the performance of the MSM, we project an input subspace $\mathcal{P}$ and reference subspaces $\mathcal{Q}$ onto the constraint subspace $\mathcal{C}$ that satisfies a constraint condition: "it includes only the essential component for recognition between class subspaces", and then the canonical angles

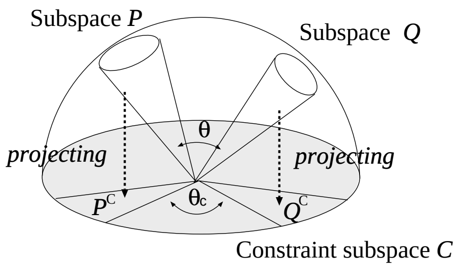

Fig. 1 Concepts of MSM and CMSM

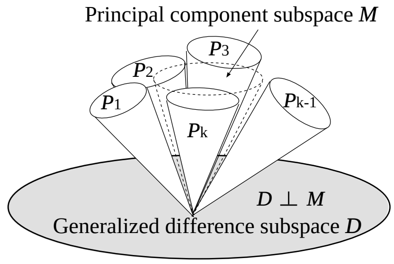

Fig. 2 Concept of generalized difference subspace

between the projected input subspace and reference subspaces as shown in Fig.1 are calculated. This MSM with the projection onto the constraint subspace $\mathcal{C}$ is called the *constrained mutual subspace method* (CMSM) [5].

3.1 Generation of constraint subspace

The constraint subspace is a generic name of the subspace which satisfies the above condition. We can use several different subspaces as constraint subspace. In this paper, we actually employ the generalized difference subspace.

The generalized different subspace is generated as follows. Given $k(\geqq 2)$ N-dimensional subspaces, the generalized difference subspace $\mathcal{D}$ is defined as the subspace which results by removing the principal component subspace $\mathcal{M}$ of all subspaces from the sum subspace $\mathcal{S}$ of these subspaces as shown in Fig.2. According to this definition, $\mathcal{D}$ is spanned by N_d eigenvectors $\mathbf{d}_i(i = N\times k - N_d, \ldots, N\times k)$ corresponding to the N_d smallest eigenvalue, of the matrix $\mathbf{G} = \sum_{i=1}^{k} \mathbf{P}_i$ of projection matrices $\mathbf{P}_i$, where the projection matrix $\mathbf{P}_i = \sum_{j=1}^{N} \mathbf{\Phi}_j^i \mathbf{\Phi}_j^{i\,T}$, $\mathbf{\Phi}_j^i$ is the j-th orthonormal basis vector of the i-th class subspace. The eigenvectors, $\mathbf{d}_i$ correspond to the i-th eigenvalue λ_i in descending order.

The effectiveness of the generalized difference subspace can be explained in several ways. First speaking qualitatively, the generalized difference subspace is the subspace which represents the difference among multiple $k(\geqq 2)$ subspaces as an extension of the difference subspace defined as the difference between two subspaces [5]. Therefore, the projection onto the generalized difference subspace is corresponding to extracting selectively the difference between subspaces. Finally, the projection onto a generalized difference subspace enables CMSM to have a higher classification ability besides the ability to tolerate variations in the face patterns.

From the other viewpoint, the projection onto the generalized difference subspace $\mathcal{D}$ corresponds to removing the principal (common) component subspace $\mathcal{M}$ from the sum subspace $\mathcal{S}$. Therefore, this projection has the effect of expanding the canonical angles between subspaces and forms a relation between subspaces which is close to the orthogonal relation, thus improving the performance of classification based on canonical angles [5]. This orthogonal process is useful for the methods based on the canonical angles. Since the orthogonal degree between the projected subspaces changes with the dimension n_c, a proper dimension has to be set by the pre-experiments.

3.2 Process flow of CMSM

The flow of CMSM is as follows. In the learning step, all the reference subspaces are generated from learning patterns using KL expansion or Gram-Schmidt method, and they are projected onto the generalized difference subspace.

The subspace $\mathcal{P}$ is projected onto the generalized difference subspace $\mathcal{D}$ by the following steps:

1. N orthogonal basis vectors of the subspace $\mathcal{P}$ are projected onto the generalized difference $\mathcal{D}$.
2. The length of each projected vector is normalized.
3. Gram-Schmidt orthogonalization is applied to the normalized vectors to obtain N orthogonal basis vectors of the subspace $\mathcal{P}^D$.

In the recognition step, the input subspace $\mathcal{P}_{in}$ is generated from input image patterns. Then, the input subspace $\mathcal{P}_{in}$ is projected onto the generalized difference subspace $\mathcal{D}$. Then, we compute the canonical angles between the projected input subspace $\mathcal{P}_{in}^D$ and the projected reference subspace $\mathcal{P}_k^D$ on a database using MSM, and the registered reference subspace that has the highest similarity is determined to be that of the identified class given the similarity is above a threshold.

Instead of using the above method of calculating canonical angles, we can use the procedure as described in [5]. In this method, firstly the set of input patterns and the reference patterns are projected, and then these subspaces are generated from the projected patterns.

4. Orthogonal mutual subspace method

The essence of OMSM is to carry out MSM using the class subspaces orthogonalized with each other in advance. We can orthogonalize class subspaces using the framework of the orthogonalization method proposed by Fukunaga & Koontz [19], [20]. This was used for the orthogonal subspace method (OSM). In this method the orthogonalization is realized by applying a transformation matrix for whitening. OMSM also utilizes this framework to orthogonalize class subspaces. However, strictly speaking, the whitening matrix used in the OMSM differs from that used in the classical OSM.

The whitening matrix used in OSM is calculated from the autocorrelation matrix of the mixture of all the classes. On the other hand, in the case of OMSM, the whitening matrix is calculated from the sum matrix of all the projection matrixes of the class subspaces.

4. 1 Calculation of the whitening matrix

In this section, we will describe how to calculate the whitening matrix $\mathbf{O}$. In the following, we will consider the case that r m-dimensional class subspaces are orthogonalized in f-dimensional space $\mathcal{I}$.

At first, we define the f×f sum matrix $\mathbf{G}$ of the r projection matrixes by the following equation.

$$\mathbf{G} = \sum_{i=1}^{r} \mathbf{P}_i \ , \tag{3}$$

where the $f \times f$ matrix $\mathbf{P}_i$ represents the projection matrix which is corresponding to the projection onto the subspace $\mathcal{P}_i$ of the class i. Using the eigenvectors and the eigenvalues of the sum matrix $\mathbf{G}$, the $v \times f$ whitening matrix $\mathbf{O}$ is defined by the following equation.

$$\mathbf{O} = \mathbf{\Lambda}^{-1/2}\mathbf{H}^T \ , \tag{4}$$

where $v = r \times m$ $(v = f,\ if\ v > f)$, and $\mathbf{\Lambda}$ is the $v \times v$ diagonal matrix with the i-th highest eigenvalue of the matrix $\mathbf{G}$ as the i-th diagonal component, and $\mathbf{H}$ is the $f \times v$ matrix whose i-th column vector is the eigenvector of the matrix $\mathbf{G}$ corresponding to the i-th highest eigenvalue. The suffix "T" represents transposition of a matrix.

We can confirm that the matrix $\mathbf{O}$ can whiten the matrix $\mathbf{G}$ by the following transformations of equation.

$$\mathbf{OGO}^T = \mathbf{\Lambda}^{-1/2}\mathbf{B}^T\mathbf{GB\Lambda}^{-1/2} \tag{5}$$

$$= \mathbf{\Lambda}^{-1/2}\mathbf{B}^T\mathbf{B\Lambda B}^T\mathbf{B\Lambda}^{-1/2} \tag{6}$$

$$= \mathbf{I}_{v \times v} \ , \tag{7}$$

where the matrix $\mathbf{I}_{v \times v}$ is a $v \times v$ unit matrix. In the above transformation, we used the relation that $\mathbf{G} = \mathbf{B\Lambda B}^T$.

The process flow of OMSM based on the whitening matrix is as follows. First, all the basis vectors of an input subspace are orthogonalized by applying the whitening matrix $\mathbf{O}$. Then, the transformed basis vectors are orthogonalized again by Gram-Schmidt orthogonalization to form the orthogonalized input subspace. One should note that if the multiple of the number r of classes and the dimension m of each class is smaller than the dimension f of feature space, the Gram-Schmidt orthogonalization method is not required, as transformed basis vectors are orthogonal to each other. Similarly, all the basis vectors of reference subspaces are orthogonalized and orthogonalized reference subspaces are generated from the transformed basis vectors. Finally, MSM is applied to the orthogonalized input subspace and the orthogonalized reference subspaces.

5. Kernel PCA

In order to understand the nonlinear extension of the linear method mentioned before, we review kernel Principal Component Analysis (KPCA) in this section.

The nonlinear function ϕ maps the patterns $\mathbf{x} = (x_1, \ldots, x_f)^\top$ of an f-dimensional input space $\mathcal{I}$ onto an f_ϕ-dimensional feature space $\mathcal{F}$: $\phi : R^f \rightarrow R^{f_\phi},\ \mathbf{x} \rightarrow \phi(\mathbf{x})$. To perform PCA on the nonlinear mapped patterns, we need to calculate the inner product $(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$ between the function values. However, this calculation is difficult, because the dimension of the feature space $\mathcal{F}$ can be very large, possibly infinite. However, if the nonlinear map ϕ is defined through a kernel function $k(\mathbf{x}, \mathbf{y})$ which satisfies Mercer's conditions, the inner products $(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$ can be calculated from the inner products $(\mathbf{x} \cdot \mathbf{y})$. This technique is known as the "kernel trick". A common choice is to use Gaussina kernel function [8]:

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{\sigma^2}\right) \ . \tag{8}$$

The function ϕ with the above exponential kernel function maps an input pattern onto an infinite feature space $\mathcal{F}$. The PCA of the mapped patterns is called kernel PCA [8]. We should note that a linear subspace generated by the kernel PCA can be regarded as a nonlinear subspace in the input space $\mathcal{I}$.

Given the N-dimensional nonlinear subspace $\mathcal{V}_k$ of class k generated from m training patterns $\mathbf{x}_i, (i = 1, \ldots, m)$, the N orthonormal basis vectors $\mathbf{e}_i^k, (i = 1, \ldots, N)$, which span the nonlinear subspace $\mathcal{V}_k$, can be represented by the linear combination of the m $\phi(\mathbf{x}_i^k), (i = 1, \ldots, m)$ as follows

$$\mathbf{e}_i^k = \sum_{j=1}^{m} \mathrm{a}_{ij}^k\, \phi(\mathbf{x}_j^k) \ , \tag{9}$$

where the coefficient a_{ij}^k is the j-th component of the eigenvector $\mathbf{a}_i$ corresponding to the i-th largest eigenvalue λ_i of the $m \times m$ kernel matrix $\mathbf{K}$ defined by the following equations:

$$\mathbf{Ka} = \lambda \mathbf{a} \tag{10}$$

$$\mathrm{k}_{ij} = (\phi(\mathbf{x}_i^k) \cdot \phi(\mathbf{x}_j^K))$$

$$= k(\mathbf{x}_i^k, \mathbf{x}_j^k) \ ,$$

where $\mathbf{a}_i$ is normalized to satisfy $\lambda_i(\mathbf{a}_i \cdot \mathbf{a}_i)$=1. We can compute the projection of the mapped $\phi(\mathbf{x})$ onto the i-th orthonormal basis vector $\mathbf{e}_i^k$ of the nonlinear subspace $\mathcal{V}_k$ of class k by the following equation:

$$(\phi(\mathbf{x}), \mathbf{e}_i^k) = \sum_{j=1}^{m} \mathrm{a}_{ij}^k \, k(\mathbf{x}, \mathbf{x}_j^k). \tag{11}$$

6. Kernel constrained mutual subspace method

6.1 Generation of the kernel difference subspace

It is possible to compute the projection of a mapped pattern $\phi(\mathbf{x})$ onto the kernel generalized difference subspace $\mathcal{D}^\phi$ using the kernel trick, since it consists of the inner products in the feature space $\mathcal{F}$. Let the N_d^ϕ-dimensional $\mathcal{D}^\phi$ be generated from the r N-dimensional nonlinear subspace $\mathcal{V}_k, (k = 1, \ldots, r)$. Firstly we calculate the orthonormal bases of kernel generalized difference subspace from all the orthonormal basis vectors of r nonlinear subspaces, namely, $r{\times}N$ basis vectors. This calculation corresponds to the PCA of all basis vectors. Define $\mathbf{E}$ to be a matrix, which contains all basis vectors as columns such as:

$$\mathbf{E} = [\mathbf{e}_1^1, \ldots, \mathbf{e}_N^1, \ldots, \mathbf{e}_1^r, \ldots, \mathbf{e}_N^r]. \tag{12}$$

Secondly, we solve the eigenvalue problem of the matrix $\mathbf{D}$ defined by the following equations.

$$\mathbf{D}\mathbf{b} = \beta\mathbf{b} \tag{13}$$

$$\mathrm{D}_{ij} = (\mathbf{E}_i \cdot \mathbf{E}_j), \quad (i, j = 1, \ldots, r{\times}N) \tag{14}$$

Where $\mathbf{E}_i$ represents the i-th column of the matrix $\mathbf{E}$.

The inner product between the i-th orthonormal basis vector $\mathbf{e}_i^k$ of the class k subspace and the j-th orthonormal basis vector $\mathbf{e}_j^{k^*}$ of the class k^* subspace can be obtained as the linear combination of kernel functions $k(\mathbf{x}^k, \mathbf{x}^{k^*})$ as follows:

$$(\mathbf{e}_i^k \cdot \mathbf{e}_j^{k^*}) = (\sum_{s=1}^{m} \mathrm{a}_{is}^k \phi(\mathbf{x}_s) \cdot \sum_{t=1}^{m} \mathrm{a}_{jt}^{k^*} \phi(\mathbf{x}_t^*)) \tag{15}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} (\phi(\mathbf{x}_s) \cdot \phi(\mathbf{x}_t^*)) \tag{16}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} \, k(\mathbf{x}_s, \mathbf{x}_t^*) \tag{17}$$

The i-th orthonormal basis vector $\mathbf{d}_i^\phi$ of the kernel generalized difference subspace $\mathcal{D}^\phi$ can be represented by a linear combination of the vectors $\mathbf{E}_j$ $(j = 1, \ldots, r{\times}N)$.

$$\mathbf{d}_i^\phi = \sum_{j=1}^{r \times N} \mathrm{b}_{ij} \mathbf{E}_j \tag{18}$$

Where the coefficient b_{ij} is j-th component of the eigenvector $\mathbf{b}_i$ corresponding to the i-th smallest eigenvalue β_i of matrix $\mathbf{D}$ under the condition that the vector $\mathbf{b}_i$ is normalized to satisfy that $\beta_i(\mathbf{b}_i \cdot \mathbf{b}_i)$=1.

Let $\mathbf{E}_j$ denote the $\eta(j)$-th basic vector of class $\zeta(j)$. The above equation is converted to the following equation:

$$\sum_{j=1}^{r \times N} b_{ij} \mathbf{E}_j = \sum_{j=1}^{r \times N} \mathrm{b}_{ij} \sum_{s=1}^{m} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \tag{19}$$

$$= \sum_{j=1}^{r \times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \tag{20}$$

6.2 Projection onto the kernel difference subspace

Although it is impossible to calculate the orthonormal basis vector $\mathbf{d}_i^\phi$ of the kernel generalized difference subspace $\mathcal{D}^\phi$, the projection of the mapped pattern $\phi(\mathbf{x})$ onto this vector $\mathbf{d}_i^\phi$ can be calculated from an input pattern $\mathbf{x}$ and all $m{\times}r$ training patterns $\mathbf{x}_s^k (s = 1, \ldots, m, k = 1, \ldots, r)$.

$$(\phi(\mathbf{x}) \cdot \mathbf{d}_i^\phi) = \sum_{j=1}^{r \times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} (\phi(\mathbf{x}_s^{\zeta(j)}) \cdot \phi(\mathbf{x})) \tag{21}$$

$$= \sum_{j=1}^{r \times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} k(\mathbf{x}_s^{\zeta(j)}, \mathbf{x}) \tag{22}$$

Note that we can compute $k(\mathbf{x}_s^{\zeta(j)}, \mathbf{x})$ through Eq.8 easily. Finally, each component of the projection $\tau(\phi(\mathbf{x}))$ of the mapped $\phi(\mathbf{x})$ onto the $N_d^\phi (< r{\times}N)$-dimensional kernel generalized difference subspace is represented as the following.

$$\tau(\phi(\mathbf{x})) = (z_1, \ z_2, \ldots, \ z_{N_d^\phi})^\top, \ z_i = (\phi(\mathbf{x}) \cdot \mathbf{d}_i^\phi) \tag{23}$$

6.3 Algorithm of KCMSM

We construct KCMSM by applying linear MSM to the projection $\tau(\phi(\mathbf{x}))$. Fig.3 shows a schematic of the KCMSM algorithm.

In the training stage, the mapped patterns $\phi(\mathbf{x}_{ki})$ of the patterns $\mathbf{x}_i^k$, $(i = 1, \ldots, m)$ belonging to class k, are projected onto the kernel difference subspace $\mathcal{D}^\phi$. Then, the N_ϕ-dimensional linear reference subspace $\mathcal{P}_k^{D^\phi}$ of each class k is generated from the mapped patterns $\tau(\phi(\mathbf{x}_i^k))$ using PCA.

In the recognition stage, we generate the linear input subspace $\mathcal{P}_{in}^{D^\phi}$ on the $\mathcal{D}^\phi$ from the input patterns $\mathbf{x}_i$,$(i = 1, \ldots, m)$. Then we compute the similarity S, defined in Sec.2.3, between the input subspace $\mathcal{P}_{in}^{D^\phi}$ and each reference subspace $\mathcal{P}_k^{D^\phi}$. Finally the reference subspace with the highest similarity S is determined to be that of the identified object, given the similarity S is above a threshold.

7. Kernel orthogonal mutual subspace method

7.1 Generation of kernel whitening matrix

The procedure of generating a nonlinear kernel whitening matrix $\mathbf{O}_\phi$ is almost the same as that of the linear whitening matrix $\mathbf{O}$. Actually $\mathbf{O}_\phi$ can be generated by only replacing the linear subspaces with nonlinear subspaces in the calculation.

In the following, we will explain how to generate the

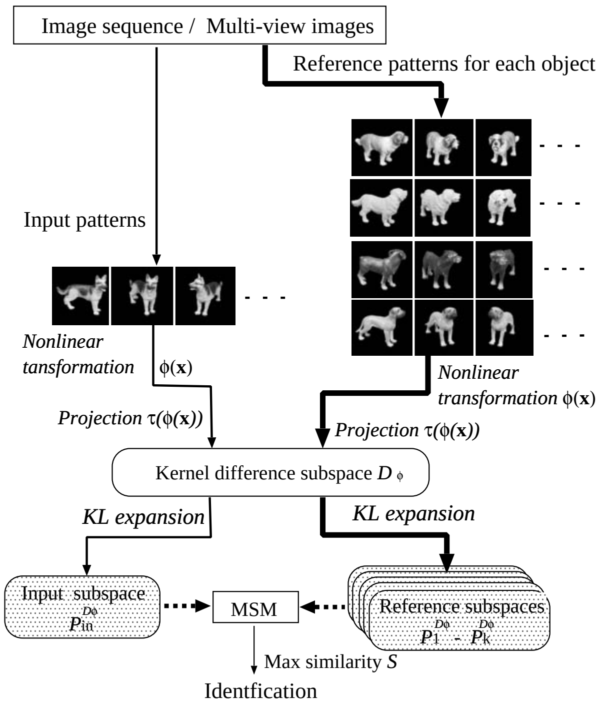

Fig. 3 Flow of object recognition using KCMSM

$n_k \times n_k$ whitening matrix $\mathbf{O}_\phi$ from all the basis vectors of r d-dimensional nonlinear class subspaces $\mathcal{V}_k (k = 1 \sim r)$, that is, $r \times d$ basis vectors. This calculation coincides to the PCA for the basis vectors of all the classes.

Assume that a class k nonlinear subspace $\mathcal{V}_k$ is generated from m learning patterns $\mathbf{x}_i^k (i = 1 \sim m)$. The d basis vectors $\mathbf{e}_i^k (i = 1 \sim d)$, which expand the subspace $\mathcal{V}_k$, are defined by the following equations.

$$\mathbf{e}_i^k = \sum_{j=1}^{m} \mathrm{a}_{ij}^k \phi(\mathbf{x}_j^k) \ , \tag{24}$$

where a_{ij}^k is the coefficient shown in the equation (11). Similarly the other nonlinear class subspaces are generated.

Next, assume that $\mathbf{E}$ is the matrix where all basis vectors are arranged as the column component.

$$\mathbf{E} = [\mathbf{e}_1^1, \ldots, \mathbf{e}_d^1, \ldots, \mathbf{e}_1^r, \ldots, \mathbf{e}_d^r] \ . \tag{25}$$

Then, we solve the eigenvalue problem of the matrix $\mathbf{Q}$ defined by the following equations.

$$\mathbf{Q}\mathbf{b} = \beta \mathbf{b} \tag{26}$$

$$\mathrm{Q}_{ij} = (\mathbf{E}_i \cdot \mathbf{E}_j), \quad (i, j = 1 \sim r \times d) \ , \tag{27}$$

where $\mathbf{E}_i$ means the i-th column component of the matrix $\mathbf{E}$. In the above equation, the inner product between i-th basis vector $\mathbf{e}_i^k$ of the class k and j-th basis vector $\mathbf{e}_j^{k^*}$ of the class k^* can be actually calculated as the linear combination of a kernel function value $k(\mathbf{x}^k, \mathbf{x}^{k^*})$ of $\mathbf{x}^k$ and $\mathbf{x}^{k^*}$.

$$(\mathbf{e}_i^k \cdot \mathbf{e}_j^{k^*}) = (\sum_{s=1}^{m} \mathrm{a}_{is}^k \phi(\mathbf{x}_s^k) \cdot \sum_{t=1}^{m} \mathrm{a}_{jt}^{k^*} \phi(\mathbf{x}_t^{k^*})) \tag{28}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} (\phi(\mathbf{x}_s^k) \cdot \phi(\mathbf{x}_t^{k^*})) \tag{29}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} k(\mathbf{x}_s^k, \mathbf{x}_t^{k^*}) \ . \tag{30}$$

The i-th row vector $\mathbf{O}_{\phi_i}$ of the kernel whitening matrix $\mathbf{O}_\phi$ can be represented as the linear combination of the basis vectors $\mathbf{E}_j$ $(j = 1 \sim r \times d)$ using the eigenvector $\mathbf{b}_i$ corresponding to the eigenvalue β_i as the combination coefficient.

$$\mathbf{O}_{\phi_i}^t = \sum_{j=1}^{r \times d} \frac{\mathrm{b}_{ij}}{\sqrt{\beta_i}} \mathbf{E}_j \ , \tag{31}$$

where the vector $\mathbf{b}_i$ is canonized to satisfy that $\beta_i(\mathbf{b}_i \cdot \mathbf{b}_i)$ is equal to 1.0. Moreover, assume that $\mathbf{E}_j$ is the $\eta(j)$-th basis vector of the class $\zeta(j)$. Then the above equation can be changed as follows.

$$\mathbf{O}_{\phi_i} = \sum_{j=1}^{r \times d} \frac{\mathrm{b}_{ij}}{\sqrt{\beta_i}} \sum_{s=1}^{m} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \tag{32}$$

$$= \sum_{j=1}^{r \times d} \sum_{s=1}^{m} \frac{\mathrm{b}_{ij}}{\sqrt{\beta_i}} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \ . \tag{33}$$

Although this vector $\mathbf{O}_{\phi_i}$ cannot be calculated practically, the inner product $(\phi(\mathbf{x}) \cdot \mathbf{O}_{\phi_i})$ with the mapped vector $\phi(\mathbf{x})$ can be calculated.

7.2 Whitening the mapped patterns

We transform the mapped vector $\phi(\mathbf{x})$ by the kernel whitening matrix $\mathbf{O}_\phi$ as the follows. This transformation can be calculated using an input vector $\mathbf{x}$ and all $r \times m$ learning vectors $\mathbf{x}_s^k (s = 1 \sim m, k = 1 \sim r)$ using the following equations.

$$(\phi(\mathbf{x}) \cdot \mathbf{O}_{\phi_i}) = \sum_{j=1}^{r \times d} \sum_{s=1}^{m} \frac{\mathrm{b}_{ij}}{\sqrt{\beta_i}} \mathrm{a}_{\eta(j)s}^{\zeta(j)} (\phi(\mathbf{x}) \cdot \phi(\mathbf{x}_s^{\zeta(j)})) \tag{34}$$

$$= \sum_{j=1}^{r \times d} \sum_{s=1}^{m} \frac{\mathrm{b}_{ij}}{\sqrt{\beta_i}} \mathrm{a}_{\eta(j)s}^{\zeta(j)} k(\mathbf{x}, \mathbf{x}_s^{\zeta(j)}) \ . \tag{35}$$

Finally, each component of the whitening transformed vector $\chi(\phi(\mathbf{x}))$ of the mapped vector $\phi(\mathbf{x})$ is represented as follows.

$$\chi(\phi(\mathbf{x})) = (z_1, \ z_2, \ldots, \ z_{r \times d})^\top \tag{36}$$

$$z_i = (\phi(\mathbf{x}) \cdot \mathbf{O}_{\phi_i})$$

7.3 Algorithm of KOMSM

We construct the nonlinear kernel orthogonal mutual subspace method (KOMSM) by applying the linear MSM to the linear subspaces generated from the whitening transformed patterns $\{\chi(\phi(\mathbf{x}))\}$. The process flow of KOMSM is as follows.

In the learning stage:

(1) The nonlinear mapped $\phi(\mathbf{x}_i^k)$ of all the patterns $\mathbf{x}_i^k (i = 1 \sim m)$ belonging to class k are transformed by the kernel whitening matrix $\mathbf{O}_\phi$.

(2) The basis vectors of the n-dimensional nonlinear orthogonal k reference subspace $\mathcal{P}_k^{D_\phi}$ are obtained as the eigenvectors of the autocorrelation matrix generated from the set of the whitening transformed pattern set $\{\chi(\phi(\mathbf{x}_1^k)), \ldots, \chi(\phi(\mathbf{x}_m^k))\}$, corresponding to the n highest values.

(3) Similarly the other nonlinear orthogonal reference subspaces are generated on the nonlinear feature space $\mathcal{F}$.

In the recognition stage:

(1) The nonlinear orthogonal input subspace $\mathcal{P}_{in}^{D_\phi}$ is also generated from the set of the whitening transformed pattern set $\{\chi(\phi(\mathbf{x}_1^{in})), \ldots, \chi(\phi(\mathbf{x}_m^{in}))\}$.

(2) The canonical angles between the input nonlinear orthogonal subspace $\mathcal{P}_{in}^{D_\phi}$ and the nonlinear reference orthogonal subspaces $\mathcal{P}_k^{D_\phi}$ are calculated as the similarity.

(3) Finally the object class is determined as the nonlinear reference orthogonal subspace with the highest similarity S, given that S is above a threshold value.

In the above process, it is possible to replace the processes of generating the nonlinear orthogonal subspaces with the following processes. Firstly the nonlinear input subspace and the nonlinear reference subspaces are generated from the set of the nonlinear mapped patterns. Next the basis vectors of the generated subspaces are transformed by the kernel whitening matrix. Finally the whitening transformed basis vectors are orthogonalized by the Gram-Schmidt method so that they span a nonlinear subspace.

8. Evaluation experiments

We compared the performances of MSM, CMSM, OMSM, KMSM, KCMSM, and KOMSM using the open data base of the multi-view image (Cropped-close128 of ETH-80) [18] and the data set of front face images collected by our self.

8.1 3D object recognition (Experiment-I)

We selected 30 similar models from the ETH-80 database as shown in Fig.4 and used them for the evaluation. The images of each model were captured from 41 views as shown in Fig.5. The view directions are the same for all models. All images are cropped, so that they contain only the object without any border area.

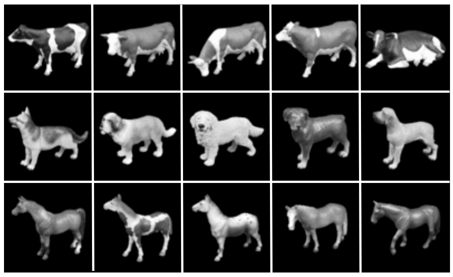

Fig. 4 Evaluation data, Top: cow, Middle: dog, Bottom: horse. This figure shows five models of 10 models.

The odd numbered images (21 frames) and the even numbered images (20 frames) were used for training and evaluation, respectively. We prepared 10 datasets for each model by making the start frame number i change from 1 to 10, where one set contains 10 frames from i-th frame to $i + 9$-th. The total number of the evaluation trials is 9000(=10×30×30).

The evaluation was performed in terms of the recognition rate and separability, which is a normalized index of classification ability. A higher value allows choosing a larger rejection threshold. The equal error rate (EER) represents the error rate at the point where the false accept rate (FAR) is equal to the false reject rate (FRR).

The experimental condition is as follows. We converted the 180×180 pixels color images to 15×15 pixels monochrome images and used them as the evaluation data. Thus, the dimension f of a pattern is 225(=15×15). The dimensions of the input subspace and the reference subspaces were set to 7 for all the methods. Gaussian kernel function with $\sigma^2 = 0.05$ defined by Eq.(8) was used for all the kernel methods.

$\mathcal{P}_{in}^D$ and $\mathcal{P}_k^D$ were generated from the patterns projected on the generalized difference subspace. The difference subspace $\mathcal{D}$ and the whitening matrix $\mathbf{O}$ were generated from thirty 20-dimensional linear class subspaces. We varied the dimension N_d of $\mathcal{D}$ between 180 and 205 to compare the performance. The kernel difference subspace $\mathcal{D}_\phi$ and the kernel whitening matrix $\mathbf{O}_\phi$ were generated from thirty 20-dimensional nonlinear class subspaces. We varied the dimension N_d^ϕ of $\mathcal{D}_\phi$ between 100 and 550.

8.2 Experimental-I results

(1)Recognition rate

Table 1 and Table 2 show the recognition rate and the separability of each method respectively. The notation used in the tables is "*method type – dimension of the difference subspace*". For example, KCMSM-550 means the KCMSM with a 550-dimensional kernel difference subspace. S[t] denotes the similarity defined using the multiple eigenvalues.

The recognition of multiple view images is typically a non-

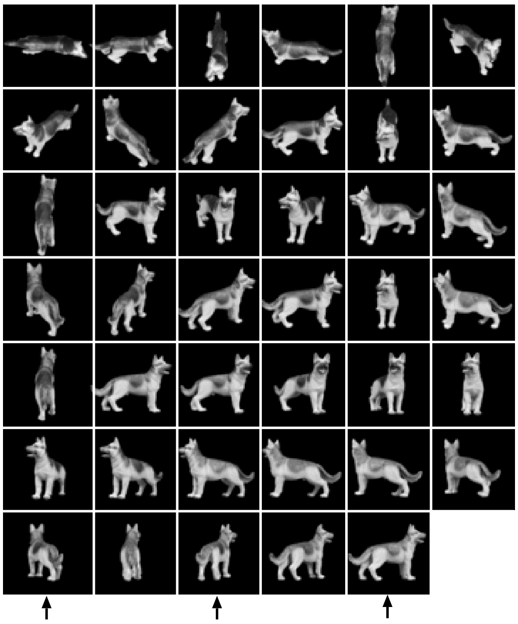

Fig. 5 All view-patterns of dog1, the row images indicated by arrow are used for learning.

Table 1 Recognition rate of each method (%), where the underlined values represent the best value (Experiment-I).

	S[1]	S[2]	S[3]	S[4]
MSM	72.7	73.7	**76.3**	74.3
CMSM-215	75.7	**81.3**	76.3	73.7
CMSM-200	73.3	81.0	79.3	77.7
CMSM-190	71.0	73.0	73.0	75.0
OMSM	51.3	54.0	**56.0**	54.0
KOMSM	85.3	87.3	88.0	**88.0**
KMSM	84.7	**87.0**	82.0	81.7
KCMSM-550	83.0	85.3	85.7	86.3
KCMSM-500	79.3	85.0	87.0	87.0
KCMSM-450	82.0	88.0	89.3	**89.7**
KCMSM-400	83.3	87.7	88.3	89.7
KCMSM-300	81.0	87.7	88.7	89.0
KCMSM-200	81.7	81.7	83.3	83.3
KCMSM-100	57.7	62.7	68.0	65.3
KCMSM-50	36.0	35.7	29.0	29.0

linear problem. This is clearly shown by the experimental results that the performance of the nonlinear methods (KMSM, KCMSM and KOMSM) is superior to that of the linear methods (MSM, CMSM and OMSM). In particular, the high separability of KCMSM and KOMSM are remarkable. This implies that they can maintain high performance even in the case of large number of classes.

The performance of MSM was improved by the nonlinear extension to KMSM where the recognition rate increased from 76.3% to 87.0% and the separability increased from 0.082 to 0.429. KCMSM improved the recognition rate further to 89.7% and increased the separability by a value of about 0.2 in comparison to KMSM. This confirms the effectiveness of the projection onto the kernel difference subspace, which serves as a feature extraction step in the feature space $\mathcal{F}$.

Table 2 Separability of each method, the underlined value means best value (Experiment-I)

	S[1]	S[2]	S[3]	S[4]
MSM	0.055	0.074	**0.082**	0.080
CMSM-215	0.203	0.236	0.242	0.236
CMSM-200	0.215	**0.257**	0.254	0.245
CMSM-190	0.229	0.255	0.249	0.244
OMSM	0.039	0.095	0.116	**0.116**
KOMSM	0.537	0.612	0.620	**0.621**
KMSM	0.375	0.420	0.420	**0.429**
KCMSM-550	0.538	0.581	0.584	0.538
KCMSM-500	0.556	0.607	0.616	0.612
KCMSM-450	0.549	0.618	0.621	**0.621**
KCMSM-400	0.529	0.601	0.607	0.609
KCMSM-300	0.483	0.536	0.545	0.545
KCMSM-200	0.340	0.385	0.403	0.408
KCMSM-100	0.141	0.194	0.212	0.213
KCMSM-50	0.057	0.080	0.085	0.089

Table 3 Recognition rate of each method (%)

Linear methods	MSM	CMSM	OMSM
EER(%)	12.0	9.5	25.0
Kernel methods	KMSM	KCMSM	KOMSM
EER(%)	6.0	4.0	4.0

KOMSM also improved the performance of OMSM largely where the recognition rate increased from 54.0% to 88.0% and the separability increases from 0.116 to 0.621. The results shows clearly the effectiveness of the orthogonalization of the class subspaces. The effectivenesses of KCMSM and KOMSM also appear in the result of ERR.

We can observe that the performance of OMSM was extremely low. This may be due to the existence of the common subspace of class subspaces. Although the matrix $\mathbf{G}$ has to have 225 positive eigenvalues if there is no common of class subspaces, it actually has only 211 positive eigenvalues.

It is very interesting that the performances of KCMSM and KOMSM are in the same level regardless of their different principles of orthogonalization. We will consider this result from the view of the degree of orthogonalization later.

The classification ability of the KCMSM and KOMSM were improved while increasing t of the similarity $S[t]$. These results show that the similarity $S[1]$ using a single image is not sufficient for classification of the models with the similar 3D shapes, as $S[1]$ utilizes the information obtained from only a single view.

(2)**Mean orthogonal degree between subspaces**

In the previous section, we have described that the performances of subspace-based methods depend on the geometrical relation between class subspaces. To confirm this, we defined the mean orthogonal degree, ω, which represents the relation between class subspaces. The value of ω is actually defined by the following equation: $\omega = 1.0 - cos^2\hat{\theta}$, where $\hat{\theta}$ is the mean value of the canonical angles θ_i between subspaces projected onto the general difference subspace or the nonlinear generalized difference subspace. The ω is closer to 1.0, when the relation between subspaces becomes closer to orthogonal relation.

Fig.6 shows the changes of the orthogonal degree ωs of CMSM and KCMSM in the case that thirty 7-dimensional class subspaces are projected onto the generalized difference subspace or the nonlinear generalized difference subspace. The different subspace and the kernel different subspace were generated from thirty 20-dimensional linear subspaces and nonlinear subspaces, respectively. In the figure, the horizontal axis represents the dimensions of the general difference subspace $\mathcal{D}$ and the kernel difference subspace $\mathcal{D}_\phi$. The vertical axis represents the mean orthogonal degree ω. The lines labeled as CMSM and KCMSM show the changes of ω of the both methods respectively.

The ω of MSM is as low as 0.160. The ω of CMSM increased to 0.725 by projecting class subspaces onto the 170-dimensional general difference subspace. On the other hand, the ω of OMSM was still as low as 0.611 despite the orthogonalization with the whitening matrix.

KOMSM obtained the maximum ω, that is, 1.0. This means that KOMSM achieved the perfect orthogonalization as expected theoretically. On the other hand, ω of KCMSM with the 450-dimensional kernel different subspace also obtained 0.991 (84.5 degree in angle).

The fact that ω of KCMSM is also almost 1.0 is corresponding well to the experimental result that KCMSM and KOMSM have the same level performance. Moreover, we can see that the dimension with the largest orthogonal degree ω is well corresponding to that with the best classification performance.

8.3 Recognition of face image (Experiment-II)

Next we conducted the evaluation experiment of all the methods using the face images of 50 persons captured under 10 kinds of lighting. We cropped the 15 ×15 pixels face images from the 320 × 240 pixels input images based on the positions of pupils and nostrils [?].

The normalized face patterns of subjects 1-25 in lighting conditions L1-L10 were used for generating the difference subspace $\mathcal{D}$, the kernel difference subspace $\mathcal{D}_\phi$, the whitening matrix $\mathbf{O}$ and the kernel whitening matrix $\mathbf{O}_\phi$. The face patterns extracted from the images of the other subjects, 26-50, in lighting conditions L1-L10 were used for evaluation.

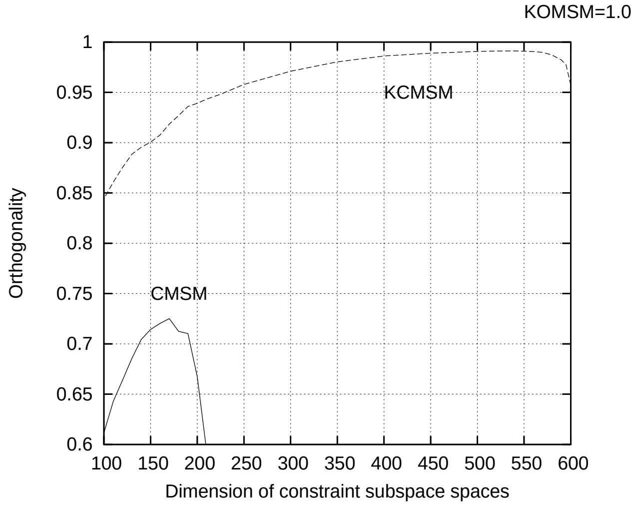

Fig. 6 Change of the mean orthogonality ω between class subspaces. The ωs of OMSM and KOMSM are 0.611 and 1.0, respectively. The ωs of MSM and KMSM are 0.160 and 0.965??, respectively.

Table 4 Recognition rate of each method (%)(Experiment-II)

Linear	MSM	CMSM-200	OMSM
Recognition rate (%)	91.74	91.3	97.09
EER(%)	12.0	7.5	6.3
Nonlinear	KMSM	KCMSM-450	KOMSM
Recognition rate (%)	91.15	97.40	97.42
EER(%)	11.0	4.3	3.5

The number of the data of each person is 150∼180 frames for each lighting condition. The data was divided into 15∼18 sub datasets by every 10 frames. The input subspace and reference subspaces were generated from these sub datasets. Finally, the total number of the evaluation trials is 18468($={}_{10}C_2 \times 25 \times 15\sim18$).

The dimension of the input subspace and reference subspaces were set to 7 for all the methods.

The difference subspace $\mathcal{D}$ and the whitening matrix $\mathbf{O}$ were generated from twenty five 60-dimensional linear subspaces of 1∼25 persons. Similarly the kernel difference subspace $\mathcal{D}_\phi$ and the kernel whitening matrix $\mathbf{O}_\phi$ were generated from twenty five 60-dimensional nonlinear class subspaces. The dimensions of the generalized difference subspace and the kernel generalized difference subspace were set to 450 and 1050, respectively. We used a Gaussian kernel with $\sigma^2 = 1.0$ for all the nonlinear methods.

The table 4 shows the recognition rate and the EER of each method. The high performances of KCMSM and KOMSM also appear in this experiment. The performance of the linear MSM is high. This means that the task of face recognition can be regarded as a linear separable problem.

9. Discussion

9. 1 Computing time

KOMSM needs to calculate all the eigenvalues and all the eigen vectors of the kernel matrix **K**. It becomes harder to calculate these, while the size of the kernel matrix **K** becomes larger depending on the number of classes. On the other hand, the calculation is easy in KCMSM since the projection of a pattern onto the general difference subspace can be indirectly calculated using only the eigenvectors corresponding to the highest eigenvalues.

9. 2 Selection of the optimal dimension of the general difference subspace

The performances of CMSM and KCMSM depend on the dimensions of the general difference subspace and the kernel difference subspace. Although the dimensions of them were selected by the pre-experiments, it is desirable that the optimal dimensions of these difference subspaces can be selected automatically in advance for real applications where the dimensions N_d^{ϕ} and N_d of the above difference subspaces are large.

From the prior discussions, it seems to be natural to consider that the optimal dimensions of the difference subspaces should be determined based on the degree of the orthogonalization between subspaces. The orthogonalization of OMSM was realized by whitening the distribution of all the basis vectors of subspaces. This leads to an idea that the distribution of the projections of all the basis vectors of subspaces on the general different subspace should be as uniform as possible in order to achieve higher orthogonalization of subspaces.

Since the distribution of the projection of the basis vectors in each direction is represented by the eigenvalue of the matrix **G**, We believe that the measuring the uniformity of all the eigenvalues is valid to select the optimal dimension in advance.

10. Conclusion

In this paper, we have compared the performances of CMSM, OMSM, and their nonlinear extensions: KCMSM and KOMSM. We confirmed that the effectivenesses of these methods are commonly derived from the orthogonalization of subspaces by the experimental result that the dimension with the largest orthogonal degree was well corresponding to that with the best classification performance. The evaluation experiments showed that KCMSM and KOMSM have the high recognition performance compared with CMSM and OMSM, and the performances of KCMSM and KOMSM are almost in the same level in spite of having different principle of orthogonailization. Since, however, the computing time of KCMSM is less than that of KOMSM, KCMSM has the advantage over KOMSM for practical applications with many objects to be classified.

References

[1] S. Watanabe, N. Pakvasa, "Subspace method of pattern recognition," Proc. 1st Int. J. Conf. on Pattern Recognition, 1973.

[2] T. Iijima, H. Genchi, K. Mori, "A theory of character recognition by pattern matching method," Proc. of 1st Int. J. Conf. on Pattern Recognition, pp.50-56, 1973.

[3] E. Oja, "Subspace methods of pattern recognition," Research Studies Press, England, 1983.

[4] O. Yamaguchi, K. Fukui, K. Maeda, "Face recognition using temporal image sequence," Proc. Third International Conference on Automatic Face and Gesture Recognition, pp.318-323, 1998.

[5] K. Fukui, O. Yamaguchi, "Face recognition using multiviewpoint patterns for robot vision," 11th International Symposium of Robotics Research (ISRR'03), pp.192-201, Springer, 2003.

[6] T. Kawahara, M. Nishiyama, O. Yamaguchi, "Face recognition by orthogonal mutual subspace method,", CVIM-151, pp.17-24, 2005 (in Japanese).

[7] K. Maeda, S. Watanabe, "A pattern matching method with local structure," Trans. IEICE, Vol.J68-D, No.3, pp.345-352, 1985 (in Japanese).

[8] B. Schölkopf, A. Smoa, and K.-R. Müller,"Nonlinear principal component analysis as a kernel eigenvalue problem, " Neural Computation, Vol. 10, pp.1299-1319, 1998.

[9] L. Wolf and A. Shashua, "Kernel principal angles for classification machines with applications to image sequence interpretation", CVPR03, pp.635-642, 2003.

[10] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, T. Darrell, "Face recognition with image sets using manifold density divergence," CVPR05, vol.1, pp.581 - 588, 2005.

[11] E. Maeda, H. Murase. "Multi-category classification by kernel based nonlinear subspace method," In Proc. the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 2, pp. 1025-1028, 1999.

[12] K. Tsuda, "Subspace classifier in the hilbert space," Pattern Recognition Letters, 20:513-519, 1999.

[13] K. Fukui, B. Stenger, O. Yamaguchi, "A framework for 3D object recognition using the kernel constrained mutual subspace method", ACCV06, part-I, pp.315-324, 2006.

[14] H. Sakano, N. Mukawa, "Kernel mutual subspace method for robust facial image recognition", Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES2000) Aug. 2000.

[15] F. Chatelin, "Eigenvalues of matrices," John Wiley & Sons, Chichester,1993.

[16] K. Maeda, O. Yamaguchi, K. Fukui, "Toward 3-dimensional pattern recognition," SSPR2004 & SPR2004, p.1061-1068, 2004.

[17] M. Nishiyama, O. Yamaguchi, K. Fukui, "Face recognition with the multiple constrained mutual subspace method," AVBPA05, pp.71-80, 2005.

[18] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization,", CVPR2003, pp.409-415, 2003.

[19] K. Fukunaga, W. C. G. Koontz, "Application of the Karhunen-Loevc expansion to feature extraction and ording," IEEE Trans. Computers, Vol.C-19,pp.311-318, 1970.

[20] K. Fukunaga, "Introduction to satistical pattern recognition," Academic Press, 1972.