A framework for 3D object recognition using the kernel constrained mutual subspace method

Kazuhiro Fukui[1], Björn Stenger[2], and Osamu Yamaguchi[2]

[1] Graduate School of Systems and Information Engineering, University of Tsukuba
kfukui@cs.tsukuba.ac.jp
[2] Corporate Research and Development Center, Toshiba corporation
bjorn@eel.rdc.toshiba.co.jp, osamu1.yamaguchi@toshiba.co.jp

Abstract. This paper introduces the kernel constrained mutual subspace method (KCMSM) and provides a new framework for 3D object recognition by applying it to multiple view images. KCMSM is a kernel method for classifying a set of patterns. An input pattern $\mathbf{x}$ is mapped into the high-dimensional feature space $\mathcal{F}$ via a nonlinear function ϕ, and the mapped pattern $\phi(\mathbf{x})$ is projected onto the kernel generalized difference subspace, which represents the difference among subspaces in the feature space $\mathcal{F}$. KCMSM classifies an input set based on the canonical angles between the input subspace and a reference subspace. This subspace is generated from the mapped patterns on the kernel generalized difference subspace, using principal component analysis. This framework is similar to conventional kernel methods using canonical angles, however, the method is different in that it includes a powerful feature extraction step for the classification of the subspaces in the feature space $\mathcal{F}$ by projecting the data onto the kernel generalized difference subspace. The validity of our method is demonstrated by experiments in a 3D object recognition task using multiview images.

1 Introduction

This paper introduces the kernel constrained mutual subspace Method (KCMSM), which provides a new framework for view-based 3D object recognition.

Many view-based methods have been proposed to achieve high-performance object recognition. Of these, the mutual subspace method (MSM)[2] with the ability of handling multiple images, such as sequential images, and multiview images, is one of the most suitable and efficient methods for object recognition. Let an $n \times n$ pixel pattern be treated as a vector $\mathbf{x}$ in n^2-dimensional space (called input space $\mathcal{I}$). In MSM, the set of patterns $\{\mathbf{x}\}$ of each class is represented by a low-dimensional linear subspace using Karhunen-Loève (KL) expansion, also known as principal component analysis (PCA). The classification of a set of patterns is executed based on the canonical angles θ_i between subspaces as shown in Fig.1, where smaller angles indicate higher similarity between two subspaces.

MSM works well when the distribution of each class can be represented by a linear subspace with no overlap of the distributions. However, this representation

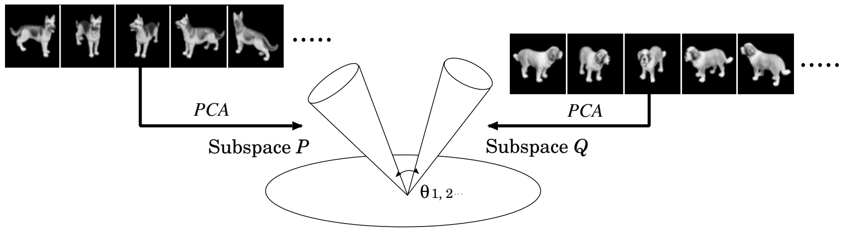


Fig. 1. Measuring the similarity between two distributions of view patterns with canonical angles $\theta_{1,2,...}$

is not suitable for representing highly nonlinear structures, such as those of multiview patterns of a 3D object.

To overcome this problem, MSM has been extended to nonlinear *kernel MSM* (KMSM)[4, 5] using the "kernel trick" [3]. An input pattern $\mathbf{x}$ is mapped onto the very high dimensional (in some cases infinite) feature space $\mathcal{F}$ via a nonlinear map ϕ. Then, MSM is applied to the linear subspaces generated from the mapped patterns $\{\phi(\mathbf{x})\}$, where the linear subspace in the feature space $\mathcal{F}$ is a nonlinear subspace as seen from the input space $\mathcal{I}$. The kernel MSM has better performance compared to MSM, since the distribution of the mapped patterns $\{\phi(\mathbf{x})\}$ can be represented by a subspace in the feature space $\mathcal{F}$ without overlapping of distributions. However, in practice the classification performance KMSM is still insufficient for many applications as is the case with other methods based on PCA, because the subspaces are generated independently of each other [1]. 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.

This issue is addressed by the *constrained MSM* (CMSM)[6]. CMSM performs the MSM algorithm on the patterns after projecting them onto the generalized difference subspace $\mathcal{D}$ (called difference subspace), wherein the differences among subspaces are contained, as shown in Fig.2. CMSM has significantly higher classification performance compared to MSM since it selectively uses the canonical angles θ_d calculated from discriminative features extracted by the projection[6, 7].

The idea in this paper is to incorporate the mechanism of this powerful feature extraction of the constrained MSM into the kernel MSM: We construct the generalized difference subspace in the feature space $\mathcal{F}$, and project the mapped pattern $\phi(\mathbf{x})$ onto this subspace for kernel MSM. This projection τ can be regarded as an effective nonlinear feature extraction step for classification of the subspaces, as seen from the input space $\mathcal{I}$. We name the difference subspace in the feature space the *nonlinear kernel generalized difference subspace* $\mathcal{D}_\phi$ and the KMSM with the projection the *kernel CMSM* (KCMSM). One question that arises is how to calculate the projection onto the difference subspace. We show that it is in fact possible to calculate the projection using the kernel trick, be-

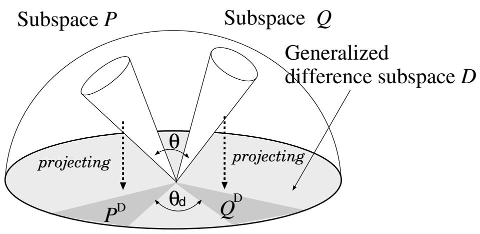


Fig. 2. Concept of CMSM

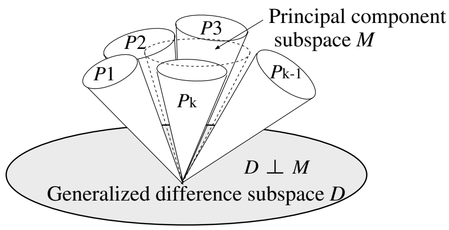


Fig. 3. Generalized difference subspace

cause it consists of the inner products. Consequently, KCMSM carries out the MSM algorithm on the extracted feature patterns $\{\tau(\phi(\mathbf{x}))\}$ by the projection.

In addition to the high classification performance, our method has also an ability of handling multiple classes in a simple framework. This is indispensable for many applications of object recognition, such as face recognition. On the other hand, many other types of kernel methods do not have this ability. For instance, the well-known support vector machine classifier is basically a two-class classifier[10]. Thus, the classification process becomes more involved and time-consuming in a multiple class problem.

This paper is organized as follows. In section 2, we review the CMSM algorithm. In section 3, we introduce the kernel generalized difference subspace in the feature space, and construct KCMSM. Our method is demonstrated by the evaluation experiments in section 4. In Section 5, conclusions are presented.

2 Recognition based on CMSM

In this section, we first review the concepts of the canonical angle and the generalized difference subspace. Then, we explain the CMSM algorithm.

2.1 Calculation of canonical angles

A natural way for comparing two subspaces is by computing the *canonical angles* between them [8]. We can obtain N canonical angles θ_i (for convenience $N \leq M$) between an M-dimensional input subspace $\mathcal{P}$ and an N-dimensional reference subspace $\mathcal{Q}$ in the f-dimensional input space $\mathcal{I}$. 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 $\mathbf{X}$[6, 8]:

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

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

2.2 Generation of the generalized difference subspace

The generalized difference subspace represents the difference among multiple $k(\geq 2)$ subspaces as an extension of the difference subspace defined as the difference between two subspaces.

Given $k(\geq 2)$ 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.3. 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 eigenvalues, of the matrix $\mathbf{G} = \sum_{i=1}^{k} \mathbf{P}_i$ of projection matrices $\mathbf{P}_i$. where the projection matrix $\mathbf{P}_i = \sum_{j=1}^{N} \mathbf{\Phi}_j^i \mathbf{\Phi}_j^{i\top}$, $\mathbf{\Phi}_j^i$ 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 projection onto the generalized difference subspace $\mathcal{D}$ corresponds to removing the principal (common) component subspace $\mathcal{M}$ from the sum subspace $\mathcal{S}$. 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 [6].

2.3 The CMSM Algorithm

The steps of the CMSM algorithm are as follows:

0. The reference subspace $\mathcal{P}_k^D$ of each class k is generated from the training patterns projected onto the generalized difference subspace $\mathcal{D}$ using PCA.
1. The input subspace $\mathcal{P}_{in}^D$ is generated from the input test patterns projected onto $\mathcal{D}$ using PCA.
2. The canonical angles θ between the $\mathcal{P}_{in}^D$ and the $\mathcal{P}_k^D$ of each class are calculated using Eq.(1).
3. The similarity $S[t]$ is calculated as the mean value $\frac{1}{t}\sum_{i=1}^{t}\cos^2\theta_i$. The reference subspace with the highest similarity is determined to be that of the identified class, given the similarity is above a threshold.

Instead of steps 0 and 1, we can also obtain the canonical angles by the procedure described in [6]. In this method, the input subspace and the reference subspaces are first generated from the set of patterns, and then these generated subspaces are projected onto $\mathcal{D}$.

3 The Kernel Constrained Mutual Subspace Method

In this section, we first review kernel Principal Component Analysis (KPCA). Next, we define the kernel generalized difference subspace using the technique of the kernel PCA, and we describe the new KCMSM algorithm.

3.1 Kernel PCA

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 $\mathcal{F}$: $\phi : R^f \to R^{f_\phi},\ \mathbf{x} \to \phi(\mathbf{x})$. To perform PCA on the 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 high, 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 the Gaussian kernel function[3]:

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

The function ϕ maps an input pattern onto an infinite feature space $\mathcal{F}$. The PCA of the mapped patterns is called kernel PCA[3], and the linear subspace generated by the kernel PCA are nonlinear subspaces 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

$$\mathbf{e}_i^k = \sum_{j=1}^{m} \mathrm{a}_{ij}^k\, \phi(\mathbf{x}_j^k), \tag{3}$$

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

$$\begin{aligned} \mathbf{K}\mathbf{a} &= \lambda \mathbf{a} \\ \mathrm{k}_{ij} &= (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)) \\ &= k(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \tag{4}$$

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 of class k by the following equation:

$$(\phi(\mathbf{x}), \mathbf{e}_i^k) = \sum_{j=1}^{m} \mathrm{a}_{ij}^k\, k(\mathbf{x}, \mathbf{x}_j). \tag{5}$$

3.2 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:

$$\mathbf{E} = [\mathbf{e}_1^1, \ldots, \mathbf{e}_N^1, \ldots, \mathbf{e}_1^r, \ldots, \mathbf{e}_N^r]. \tag{6}$$

Secondly we solve the eigenvalue problem of the matrix $\mathbf{D}$ defined by the following equation:

$$\mathbf{D}\mathbf{b} = \beta\mathbf{b} \tag{7}$$

$$\mathrm{D}_{ij} = (\mathbf{E}[i] \cdot \mathbf{E}[j]), \quad (i, j = 1, \ldots, r \times N)\ , \tag{8}$$

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:

$$(\mathbf{e}_i^k \cdot \mathbf{e}_j^{k^*}) = (\sum_{s=1}^{m} \mathrm{a}_{is}^k \phi(\mathbf{x}_s) \cdot \sum_{t=1}^{m} \mathrm{a}_{jt}^{k^*} \phi(\mathbf{x}_t^*)) \tag{9}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} (\phi(\mathbf{x}_s) \cdot \phi(\mathbf{x}_t^*)) \tag{10}$$

$$= \sum_{s=1}^{m} \sum_{t=1}^{m} \mathrm{a}_{is}^k \mathrm{a}_{jt}^{k^*} k(\mathbf{x}_s, \mathbf{x}_t^*) \tag{11}$$

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 $\mathbf{E}[j]$ ($j = 1, \ldots, r \times N$), $\mathbf{d}_i^\phi = \sum_{j=1}^{r \times N} \mathrm{b}_{ij} \mathbf{E}[j]$, where the weighting coefficient b_{ij} is the 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 orthonormal basis vector $\mathbf{d}_i^\phi$ is converted to the following equation:

$$\sum_{j=1}^{r \times N} \mathrm{b}_{ij} \mathbf{E}[j] = \sum_{j=1}^{r \times N} \mathrm{b}_{ij} \sum_{s=1}^{m} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)}) \tag{12}$$

$$= \sum_{j=1}^{r \times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} \phi(\mathbf{x}_s^{\zeta(j)})\ . \tag{13}$$

3.3 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

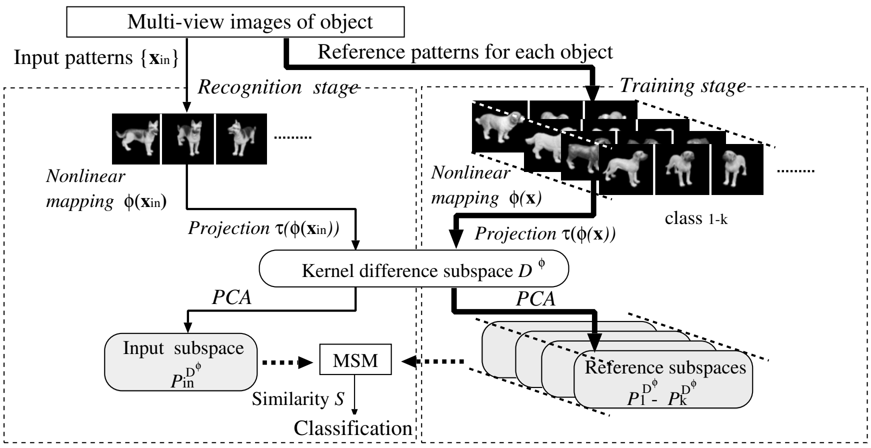


Fig. 4. Flow of object recognition using KCMSM

$\phi(\mathbf{x})$ onto this vector $\mathbf{d}_i^{\phi}$ can be calculated from an input pattern $\mathbf{x}$ and all $m \times r$ training patterns $\mathbf{x}_s^k (s = 1, \ldots, m, k = 1, \ldots, r)$.

$$(\phi(\mathbf{x}) \cdot \mathbf{d}_i^{\phi}) = \sum_{j=1}^{r \times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} (\phi(\mathbf{x}_s^{\zeta(j)}) \cdot \phi(\mathbf{x})) \tag{14}$$

$$= \sum_{j=1}^{r \times N} \sum_{s=1}^{m} \mathrm{b}_{ij} \mathrm{a}_{\eta(j)s}^{\zeta(j)} k(\mathbf{x}_s^{\zeta(j)}, \mathbf{x}) \tag{15}$$

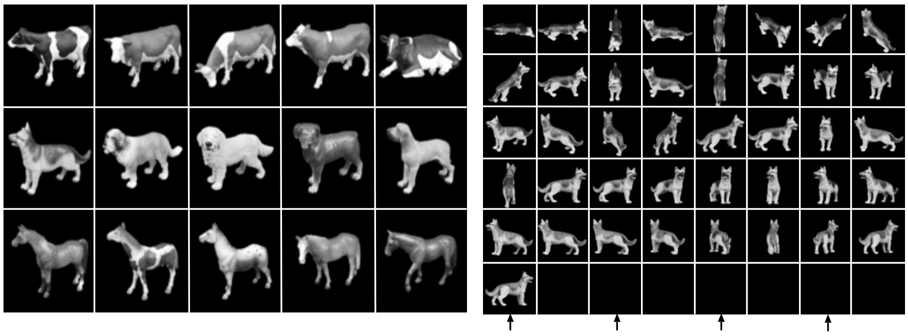
Note that we can compute $k(\mathbf{x}_s^{\zeta(j)}, \mathbf{x})$ through Eq.(2) easily. Finally, each component of the projection $\tau(\phi(\mathbf{x}))$ of the mapped $\phi(\mathbf{x})$ onto the $N_d^{\phi} (< r \times N)$-dimensional kernel generalized difference subspace is represented as the following: $\tau(\phi(\mathbf{x})) = (z_1,\ z_2, \ldots,\ z_{N_d^{\phi}})^{\top},\ z_i = (\phi(\mathbf{x}) \cdot \mathbf{d}_i^{\phi})$.

3.4 The KCMSM Algorithm

We construct KCMSM by applying linear MSM to the projection $\tau(\phi(\mathbf{x}))$. Fig.4 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 object class is determined as the reference subspace with the highest similarity S, given that S is above a threshold value.



(a) (b)

Fig. 5. Data set: (a) Subset of the input images, Top: cows, Middle: dogs, Bottom: horses. (b) All 41 view-patterns of a dog model: the columns indicated by the arrows are used as the training data

4 Evaluation experiments

We compared KCMSM with MSM, CMSM, and KMSM using the public database of the multi-view image set (ETH-80: Cropped-close128)[9].

Experimental conditions: We selected 30 similar models (10 of each; cows, dogs, and horses) from the database as shown in Fig.5(a) and used them for the evaluation. The images of each model were captured from 41 views as shown in Fig.5(b). The view directions are the same for all models. All images are cropped, so that they contain only the object without any border area.

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 10 frames from i-th frame to $i + 9$-th is one set. The total number of the evaluation trials is 9000(=10×30×30). The evaluation was performed using measures for recognition rate and separability: a normalized index of classification ability. Given two classes of similarities within a model category and similarities across different model category, separability was calculated as a ratio of the between-class scatter to the total scatter.

We converted the 180×180 pixels color images to 15×15 pixels monochrome images and use 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 in all methods.

$\mathcal{P}_{in}^D$ and $\mathcal{P}_k^D$ were generated from the patterns projected on the generalized difference subspace. The difference subspace $\mathcal{D}$ was generated from thirty 20-dimensional subspaces of all classes according to the procedure described in Sec. 2.2. We varied the dimension N_d of $\mathcal{D}$ between 190 and 215 to compare the performance. The kernel difference subspace $\mathcal{D}_\phi$ was generated from thirty 20-dimensional subspaces of all classes according to the procedure described in

Table 1. Performance of each method

(a) Recognition rate (%)

	t=1	t=2	t=3	t=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
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

(b) Separability

	t=1	t=2	t=3	t=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
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

Sec.2.3. We varied the dimension N_d^ϕ of $\mathcal{D}_\phi$ between 100 and 550. We used a Gaussian kernel with $\sigma^2 = 0.05$ defined by Eq.(2).

Experimental results: Table 1 shows the recognition rate and the separability. In the tables, the notation *method type – dimension of the difference subspace* is used and t denotes the number of the canonical angles used for the similarity S[t] defined in step 3 of Section 2.3.

From these results, it can be observed that the performance of the nonlinear methods (KMSM and KCMSM) is superior to the one of the linear methods (MSM and CMSM), indicating that the recognition of multiple view images is typically a nonlinear problem.

The performance of MSM was improved by the nonlinear extension of MSM to KMSM where the recognition rate increased from 76.3% to 87.0% and the separability increased from 0.082 to 0.429.

The new KCMSM improved the recognition rate further to 89.7% and increased the separability by a value of almost 0.2 in comparison to KMSM. This confirms the effectiveness of projection the onto the kernel difference subspace, which serves as a feature extraction step in the feature space $\mathcal{F}$. In particular, the high separability of KCMSM is remarkable. This indicates that KCMSM can maintain high performance even if the number of classes becomes larger.

The classification ability of KCMSM was improved while increasing t of the similarity $S[t]$. These results show that the similarity $S[1]$ is not sufficient for classification of the models with similar 3D shapes. This is because $S[1]$ utilizes only the information of a single view. On the other hand, $S[t](t \geq 2)$ reflects the information of 3D shape including multiple views. Note that the recognition rate of KMSM decreased, although it is also a nonlinear method. From this, one can deduce that the projection onto the kernel difference subspace ensures the validity of the similarity $S[t], (t \geq 2)$.

In comparison between KCMSM-450 and KCMSM-300, the extreme degradation of performance does not appear even when the dimension of the kernel difference subspace decreased to 300. This implies that we can decrease the dimension N_d^ϕ within the permissible range to reduce the computing cost.

5 Summary and Conclusions

This paper has introduced the kernel constrained mutual subspace method (KCMSM) and demonstrated its application to 3D object recognition. We showed a significant performance improvement over kernel MSM, which is a state-of-the-art method for classifying multiple view patterns with nonlinear structure. The projection onto the kernel generalized difference subspace can be viewed as a nonlinear feature extraction step based on the concept of constrained MSM. The extracted features by this projection could improve the classification ability of kernel MSM. The validity of KCMSM was shown through the experimental results with the set of the multiple view patterns of 3D objects.

In future work, we will evaluate the performance of KCMSM using other databases, such as a face image database. In this case, the comparisons with other kernel methods[10] are required. Another problem that remains to be addressed is the computation of the eigen–problems of the matrices **K** and **D**, which becomes difficult when the size of these matrices become large in proportion to the numbers of the classes and the training patterns. To solve this problem, the reduction of the number of the training patterns is most effective. Thus, the framework of ensemble learning[11] is useful, since it can obtain high performance using only a few training patterns.

References

1. Oja, E.: Subspace methods of pattern recognition. Research Studies Press, England, (1983)
2. Yamaguchi, O., Fukui, K., and Maeda, K.: Face recognition using temporal image sequence. Proc. Third International Conference on Automatic Face and Gesture Recognition, (1998) 318–323
3. Schölkopf, B., Smola, A. and Müller, K.-R.: Nonlinear principal component analysis as a kernel eigenvalue problem. Neural Computation, vol. 10, (1998) 1299–1319
4. Sakano, H., and Mukawa, N.: Kernel mutual subspace method for robust facial image recognition. Fourth International Conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES2000) (2000) 245–248
5. Wolf, L. and Shashua, A.: Kernel principal angles for classification machines with applications to image sequence interpretation. Proc. CVPR, (2003) 635–642
6. Fukui, K. and Yamaguchi, O.: Face recognition using multi-viewpoint patterns for robot vision. 11th International Symposium of Robotics Research (ISRR'03), Springer, (2003) 192–201
7. Arandjelović, O., Shakhnarovich, G., Fisher, J., Cipolla, R. and Darrell, T.: Face recognition with image sets using manifold density divergence. Proc. CVPR, vol.1, (2005) 581–588
8. Chatelin, F.: Eigenvalues of matrices. John Wiley & Sons, Chichester, (1993)
9. Leibe, B. and Schiele, B.: Analyzing appearance and contour based methods for object categorization. Proc. CVPR, (2003) 409–415
10. Schölkopf, B. and Smola, A.: Learning with Kernels. The MIT Press, (1999)
11. Breiman, L.: Bagging Predictors. Machine Learning, vol.24, no.2 (1996) 123–140