# Face Recognition with the Multiple Constrained Mutual Subspace Method

Masashi Nishiyama[1], Osamu Yamaguchi[1], and Kazuhiro Fukui[2]

[1] Corporate Research & Development, Toshiba Corporation, Japan
{masashi.nishiyama,osamu1.yamaguchi}@toshiba.co.jp
[2] Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
kfukui@cs.tsukuba.ac.jp

**Abstract.** In this paper, we propose a novel method named the *Multiple Constrained Mutual Subspace Method* which increases the accuracy of face recognition by introducing a framework provided by ensemble learning. In our method we represent the set of patterns as a low-dimensional subspace, and calculate the similarity between an input subspace and a reference subspace, representing learnt identity. To extract effective features for identification both subspaces are projected onto multiple constraint subspaces. For generating constraint subspaces we apply ensemble learning algorithms, i.e. Bagging and Boosting. Through experimental results we show the effectiveness of our method.

## 1 Introduction

Recently, many face identification methods that perform recognition from a set of patterns instead of a single pattern have been proposed[1–5]. Since these methods are able to cope with variation in appearance under varying pose, a robust face identification application can be built.

To identify faces using a set of patterns, we have previously proposed the *Mutual Subspace Method* (MSM)[1]. In MSM, a set of patterns is represented as a low-dimensional subspace. To compare the input subspace with the reference subspace representing learnt identity, we calculate their similarity which is defined by the minimum angle between the input subspace and the reference subspace. These subspaces are generated using principal component analysis (PCA).

To improve the performance of MSM we have extended this method to the *Constrained Mutual Subspace Method* (CMSM)[5]. In CMSM, to extract effective features for identification, we project the input subspace and the reference subspace onto the constraint subspace, as shown in Fig. 1. Through this projection we can extract features that are insensitive to varying facial pose and illumination, while remaining sensitive to change in individual appearance. Using CMSM Sato et al.[6] illustrated the effectiveness in a practical security system, while Kozakaya et al.[7] demonstrated an implementation of a real-time system on an image processing LSI chip.

T. Kanade, A. Jain, and N.K. Ratha (Eds.): AVBPA 2005, LNCS 3546, pp. 71–80, 2005.

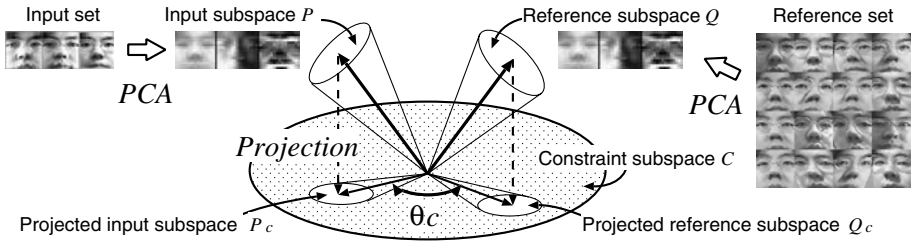



**Fig. 1.** Concept of CMSM. The input subspace and the reference subspace are generated from the set of patterns. Then, both subspaces are projected onto the constraint subspace. Finally, the similarity is determined with the angle $\theta_C$

Although CMSM is effective, a large number of training patterns are required for generating the constraint subspace. Since variation in appearance is large under varying pose and illumination, it is difficult to acquire training patterns which sufficiently represent these variations. Therefore, we need a method of generating the constraint subspace which yields high performance from a limited number of acquired training patterns. In the field of machine learning, ensemble learning has been proposed[8, 9]. Ensemble learning derives recognition performance by combining hypotheses obtained from given training samples. Wang et al.[10] applied ensemble learning to face identification based on Linear Discriminant Analysis and demonstrated that they obtain high performance using only a few training patterns.

In this paper we propose a new method which generates multiple constraint subspaces by introducing the framework provided by ensemble learning. Using these constraint subspaces, we extend CMSM to the *Multiple Constrained Mutual Subspace Method* (MCMSM). In MCMSM, the input subspace and the reference subspace are projected onto each constraint subspace, and the similarity is calculated on each constraint subspace. By combining these similarities we finally determine the combined similarity as shown in Fig. 2. To generate constraint subspaces, we propose two approaches in which we apply the framework provided by ensemble learning.

This paper is organized as follows. First, we describe the method for applying MCMSM to face identification in section 2. Next, we describe two approaches for generating constraint subspaces in section 3. Then, we demonstrate the effectiveness of our method using MCMSM by experiments in section 4.

## 2 Identification Using MCMSM

### 2.1 Algorithm for Face Identification

In this section, we describe the procedure of our face identification method. First, an input set of face patterns is obtained from a video sequence. We locate the face pattern from the positions of the pupils and the nostrils obtained automatically by the method described in [1, 7]. The pattern is transformed to a vector by

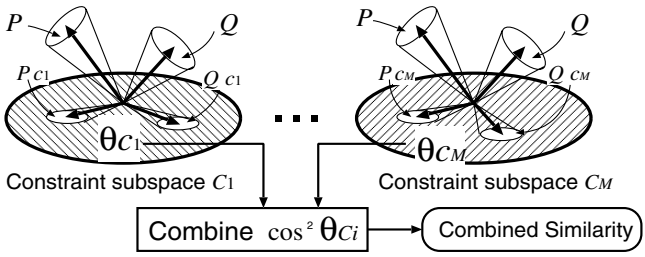



**Fig. 2.** Concept of MCMSM. The input subspace $P$ and the reference subspace $Q$ are projected onto each constraint subspace $Ci$. By combining $M$ similarities($cos^2\theta_{Ci}$), which are calculated on $Ci$, we finally determine the combined similarity

raster-scanning of the pattern, and we apply PCA to the vectors to generate an input subspace. Let $\mathbf{x}$ be a vector and $N_V$ be the number of the vectors, the basis vectors of the input subspace are the eigenvectors of the correlation matrix $\mathbf{A} = 1/N_V \sum_{i=1}^{N_V} \mathbf{x}\mathbf{x}^T$[12].

To compare the input subspace with the reference subspace, registered in a database for each individual, we calculate their combined similarity. This combined similarity is determined with similarities calculated on each constraint subspace. The identified person is determined as corresponding to the reference subspace of the highest combined similarity. The details of each process are described in the following section.

### 2.2 Projection onto Constraint Subspaces

To project the input subspace $P$ onto $M$ constraint subspaces, we carry out the following steps:

1. Project basis vectors of $P$ onto the $i$-th constraint subspace $C_i$.
2. Normalize the length of each projected vector.
3. Apply Gram-Schmidt orthogonalization to the normalized vectors.

The orthogonal normalized vectors are basis vectors of the projected input subspace $P_{C_i}$. This procedure is repeated $M$ times for each constraint subspace. The projected reference subspace $Q_{C_i}$ can be obtained with the same procedure.

### 2.3 Calculation of the Similarity on Each Constraint Subspace

We define similarity $S_{C_i}$ between the subspace $P_{C_i}$ and the subspace $Q_{C_i}$ as

$$S_{C_i} = \cos^2 \theta_{C_i} \ , \tag{1}$$

where $\theta_{C_i}$ represents the canonical angle between $P_{C_i}$ and $Q_{C_i}$. The canonical angle is calculated using MSM[1]. The similarity $S_{C_i}$ can be obtained from the largest eigenvalue $\lambda_{max}$ of $\mathbf{X}$ using

$$\mathbf{X}\mathbf{a} = \lambda \mathbf{a} \ , \tag{2}$$

$$\mathbf{X} = (x_{mn}) \quad m, n = 1 \ldots N \ , \ and \tag{3}$$

$$x_{mn} = \sum_{l=1}^{N} (\psi_m, \phi_l)(\phi_l, \psi_n) \ , \tag{4}$$

where $\psi_m$ is the $m$-th basis vector of subspace $P_{C_i}$; $\phi_l$ is the $l$-th basis vector of subspace $Q_{C_i}$; $(\psi_m, \phi_l)$ is the inner product of $\psi_m$ and $\phi_l$; $N$ is the dimension of $P_{C_i}$ and $Q_{C_i}$. The similarity $S_{C_i}$equals $\lambda_{max}$. If the input subspace and the reference subspace are identical, the canonical angle $\theta_{C_i}$ equals 0.

### 2.4 Combine Similarities

To combine similarities obtained on each constraint subspace, we define the combined similarity $S_T$ as follows:

$$S_T = \sum_{i=1}^{M} \alpha_i S_{C_i} \ , \tag{5}$$

where $M$ is the number of the constraint subspaces; $\alpha_i$ is the $i$-th coefficient of $C_i$; $S_{C_i}$ is the similarity between $P_{C_i}$ and $Q_{C_i}$ projected onto $C_i$.

## 3 Generation of Multiple Constraint Subspaces with Ensemble Learning

In this section, we explain the method of generating a single constraint subspace for CMSM[5]. Next, we describe two approaches for generating multiple constraint subspaces with ensemble learning for MCMSM.

### 3.1 Generation of a Single Constraint Subspace

To allow for the variation in appearance for each individual, we acquire the sets of patterns while changing illumination and pose for $L$ individuals. The variation of the patterns is represented as a subspace for each individual. We call this subspace the *training subspace*.

To generate a constraint subspace which separates the training subspaces by projection, we calculate eigenvectors using

$$(\mathbf{P}_1 + \mathbf{P}_2 + \ldots + \mathbf{P}_L)\mathbf{a} = \lambda \mathbf{a} \ , \tag{6}$$

$$\mathbf{P}_j = \sum_{k=1}^{N_B} \psi_{jk} \psi_{jk}^T \ , \tag{7}$$

where $\mathbf{P}_j$ is the projection matrix of the $j$-th training subspace; $N_B$ is the dimension of training subspace; $\psi_{jk}$ is the $k$-th basis vector of the $j$-th training subspace. The eigenvectors, selected in ascending order, are the basis vectors of the constraint subspace. For details of CMSM see [5, 7].

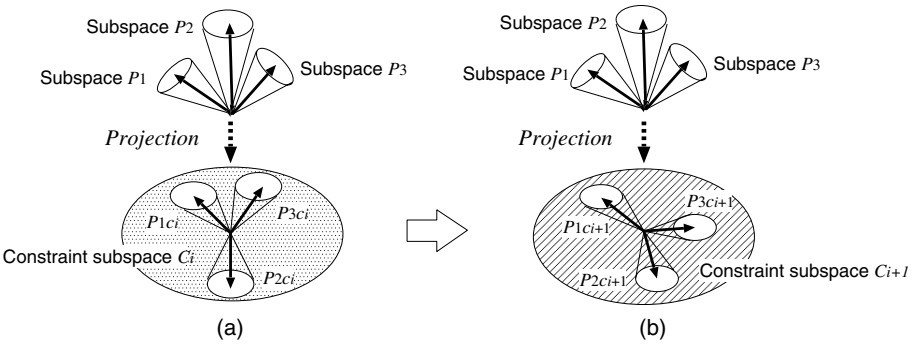



**Fig. 3.** Concept of the method for generating constraint subspaces using Boosting

### 3.2 Generation of Constraint Subspaces with Bagging

To generate constraint subspaces, we use Bagging[8], which is based on an ensemble learning algorithm. Multiple classifiers are constructed using random sampling in Bagging. To apply this framework to generating constraint subspaces, we randomly select $L'(< L)$ subspaces from $L$ training subspaces. Each constraint subspace is generated independently using selected training subspaces.

**Algorithm Using Bagging**
To summarize: we generate $M$ constraint subspaces by the following steps:

1. Select $L'$ training subspaces randomly without replacement.
2. Generate a constraint subspace using selected $L'$ training subspaces in eq.(6).
3. Until $M$ constraint subspaces are generated, go to 1.

### 3.3 Generation of Constraint Subspaces with Boosting

In another method of generating constraint subspaces, we use Boosting[9]. Each classifier is constructed sequentially by reweighting the training patterns in Boosting. The current weight is given to training patterns which were classified incorrectly in the previous constructed classifier.

In applying this framework to generate constraint subspaces we must define how to calculate the weight for each training subspace. Consider similarities between training subspaces on the constraint subspace. As shown in Fig. 3(a), when the projected training subspace $P_{1C_i}$ and the projected training subspace $P_{3C_i}$ are similar on the constraint subspace $C_i$, the likelihood of the false identification is increased for these training subspaces. To cope with this problem, we aim to separate $P_{1C_{i+1}}$ and $P_{3C_{i+1}}$ on $C_{i+1}$ as shown in Fig. 3(b). To achieve this, we generate $C_{i+1}$ by assigning large weight to $P_{1C_i}$ and $P_{3C_i}$, thereby increasing their importance and decreasing the remaining error.

**Algorithm Using Boosting**
To summarize: we generate $M$ constraint subspaces by the following steps:

1. Define the initial weight $W_1(j)$.
2. Generate the $i$-th constraint subspace $C_i$ using $i$-th weight $W_i(j)$ and the projection matrix $\mathbf{P}_j$ of the $j$-th training subspace as

$$(W_i(1)\mathbf{P}_1 + \ldots + W_i(L)\mathbf{P}_L)\mathbf{a} = \lambda\mathbf{a} \ . \tag{8}$$

3. Calculate the next weight $W_{i+1}(j)$ using $C_i$.
4. Until $M$ constraint subspaces are generated, go to 2.

The weight $W_{i+1}(j)$ is calculated using

$$W_{i+1}(j) = \frac{S'_j}{\sum_{j=1}^{L} S'_j} \quad and \tag{9}$$

$$S'_j = \sum_{j' \neq j}^{L} \beta_{jj'} \ , \tag{10}$$

where $\beta_{jj'}$ equals $\theta_{C_{ijj'}}$; $\theta_{C_{ijj'}}$ is the angle between $P_j$ and $P_{j'}$ projected onto the $C_i$. To generate a constraint subspace using only similar training subspaces, we can set threshold $T$ to be

$$\beta_{jj'} = \begin{cases} \cos^2\theta_{C_{ijj'}} & T \leq \cos^2\theta_{C_{ijj'}} \\ 0 & T > \cos^2\theta_{C_{ijj'}} \end{cases} \ . \tag{11}$$

## 4 Empirical Evaluation

### 4.1 Performance for Varying Illumination

To illustrate the performance of our face identification method, the lighting condition was changed dynamically. We collected a video sequence at 5 frames per second for each person under each lighting condition. We set 10 difference lighting conditions using 7 light sources (A-G); see Fig. 4. A image of the set of each lighting condition is shown in Fig. 5(a). A video sequence consisted of 140 face images which were captured in arbitrary facial pose, e.g. translation, yaw and pitch. The size of each image was 240×320 pixels and 50 different individuals' data were collected. From each image a $30 \times 30$ pixel pattern, as shown in Fig. 5(b), was extracted. This pattern was histogram equalized, resized to $15 \times 15$ pixels by subsampling, a vertical gradient operator was applied, and finally the pattern was transformed to a $15 \times (15-1) = 210$-dimensional vector.

We divided the data into two groups that each consisted of 25 individuals' patterns. The first group was used for identification and the second for generating constraint subspaces. In the first group, we divided the patterns into input sets and reference sets for each person. An input set consisted of 10 patterns for each lighting condition and a reference set consisted of 70 patterns for each lighting condition. We used 7 input sets per person for each lighting condition. In the second group, to learn variation of patterns under varying illuminations, a

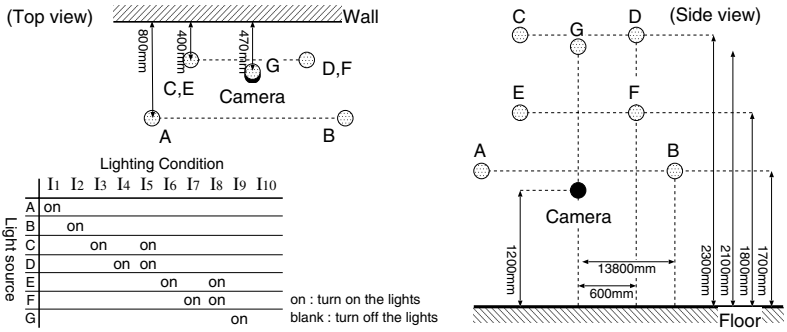
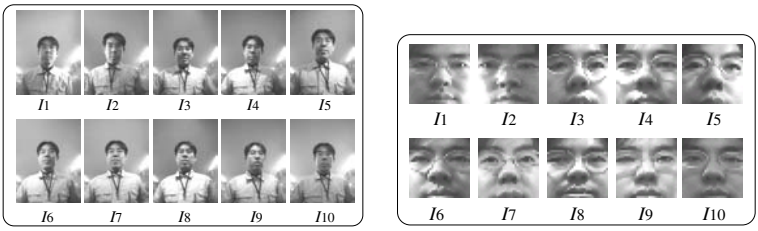



**Fig. 4.** Setting of light sources and camera




(a) captured images (b) face patterns

**Fig. 5.** Examples of the captured images and the face patterns

training subspace was generated by using all lighting condition patterns. A set of training patterns consisted of 140 patterns per lighting condition. We generated 25 training subspaces.

We compared the performance of MCMSM with those of conventional methods.

**(a)Nearest Neighbor (NN)**
The similarity was determined with the smallest Euclidean distance between the pattern in the input set and the pattern in the reference set.

**(b)Subspace Method[11] (SM)**
The similarity was determined using the average of the angle calculated between each pattern of the input set and the reference subspace. We generated the 40-dimensional reference subspace for each reference set.

**(c)Mutual Subspace Method[1] (MSM)**
The similarity was determined using the angle between the input subspace and the reference subspace. We generated the 7-dimensional input subspace for each input set and the 7-dimensional reference subspace for each reference set.

**(d)Constrained MSM[5] (CMSM)**
The similarity was determined with MSM after projection onto a single constraint subspace. The constraint subspace was generated with $L = 25$ training subspaces. We set the dimension of the training subspace to $N_B = 30$, and the dimension of the constraint subspace to $N_C = 170$.

**(e)Multiple CMSM with Bagging (MCMSM-Bagging)**
The similarity was determined with MSM after projecting onto multiple constraint subspaces. Each constraint subspace was generated from $L' = 8$ training subspaces selected randomly. We used $M = 10$ constraint subspaces. The coefficient of the combining was $\alpha_i = 1/10$.
**(f)Multiple CMSM with Boosting (MCMSM-Boosting)**
The similarity was determined with MSM after projecting onto multiple constraint subspaces. Each constraint subspace was generated from weighted training subspaces. We used $M = 10$ constraint subspaces. The initial weight $W_1(j)(j = 1 \ldots 25)$ was $1/25$, the threshold $T$ was $3.5\sigma_i$, and $\sigma_i$ was the standard deviation of similarities which were calculated between training subspaces. The coefficient $\alpha_i$ was $1/10$.

Table 1 shows the evaluation result of each method in terms of the correct match rate (CMR) and the equal error rate (EER). CMR is the probability that an input set of the right person is correctly accepted. EER is the probability that the false acceptance rate (FAR) equals the false rejection rate (FRR). We can see that the methods (e) and (f) using MCMSM are superior to (a)-(d) with regard to CMR and EER. Figure 6 shows the receiver operating characteristic (ROC) curves, which indicate FAR and FRR of each method. The superiority of MCMSM (e) and (f) is also apparent from this.

**Table 1.** Experimental results under varying illumination (25 registered persons)

| | Method | CMR(%) | EER(%) |
|---|---|---|---|
| (a) | NN | 95.4 | 23.9 |
| (b) | SM | 95.4 | 12.9 |
| (c) | MSM | 95.4 | 9.8 |
| (d) | CMSM | 95.4 | 5.0 |
| (e) | MCMSM-Bagging | 98.2 | 4.0 |
| (f) | MCMSM-Boosting | 98.6 | 3.9 |

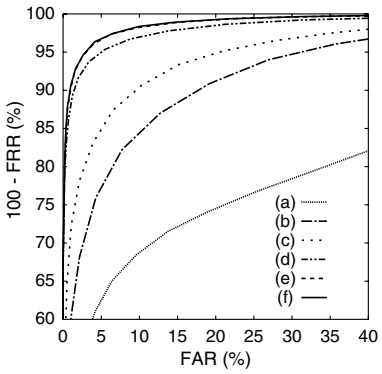



**Fig. 6.** ROC curves (25 registered persons, varying illumination)

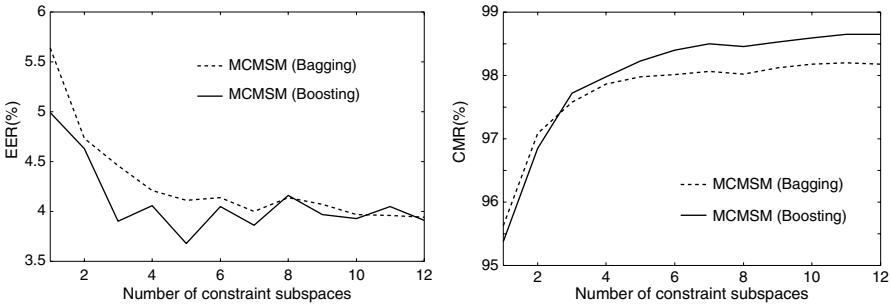



**Fig. 7.** Identification performance with increasing the number of constraint subspaces

Figure 7 shows the performance of MCMSM versus the number of constraint subspaces. We can see improved performance for both generating methods as the number of constraint subspaces increased.

### 4.2 Performance Assessment on a Large Database

To evaluate performance for a large number of individuals, we collected a total of 1000 input sets for 500 people. The facial pose changed irregularly at each input set although the lighting conditions were almost uniform. As before the dimension of the vector was 210. An input set consisted of 15 patterns and a reference set consisted of 125 patterns. We compared the performance of three methods: (i)CMSM, (ii)MCMSM-Bagging and (iii)MCMSM-Boosting. In these methods, we used the 7-dimensional input subspace and the 7-dimensional reference subspace. We used 500 training subspaces for generating the constraint subspace. The training subspace was generated with the reference set. The dimension of the training subspace was $N_B = 10$, and the dimension of the constraint subspace was $N_C = 170$. In (i), we used a single constraint subspace generated with 500 training subspaces. In (ii), we used $M = 10$ constraint subspaces. Each constraint subspace was generated from $L' = 30$ training subspaces. The coefficient $\alpha_i$ was 1/10. In (iii), we used $M = 10$ constraint subspaces. The initial weight $W_1(j)$ was 1/500, the threshold $T$ was $5\sigma_i$, and the coefficient $\alpha_i$ was 1/10.

Table 2 shows the evaluation result of each method. We can see that the methods using MCMSM are superior to that using CMSM.

**Table 2.** Experimental results (500 registered persons)

| | Method | CMR (%) | EER (%) |
|---|---|---|---|
| (i) | CMSM | 94.7 | 2.3 |
| (ii) | MCMSM-Bagging | 96.2 | 1.6 |
| (iii) | MCMSM-Boosting | 96.8 | 1.6 |

## 5 Conclusion

This paper presented the *Multiple Constrained Mutual Subspace Method* in which we applied ensemble learning to the *Constrained Mutual Subspace Method*. To extract effective features for face identification, we project the input subspace and the reference subspace onto multiple constraint subspaces. In the experiment we obtained high performance compared with projecting onto a single constraint subspace. To generate constraint subspaces, we apply the framework provided by ensemble learning, i.e. Bagging, Boosting. We evaluated the algorithms on a database of varying illumination and a database with 500 individuals. The effectiveness of MCMSM is demonstrated on both databases.

## References

1. Yamaguchi, O., Fukui, K., and Maeda, K.: Face recognition using temporal image sequence. Proceedings IEEE Third International Conference on Automatic Face and Gesture Recognition, (1998) 318-323
2. Shakhnarovich, G., Fisher, J.W., and Darrell, T.: Face Recognition from Long-Term Observations. Proceedings of European Conference on Computer Vision, (2002) 851-868
3. Wolf, L., and Shashua, A.: Learning over Sets using Kernel Principal Angles. Journal of Machine Learning Research, 4:913-931, (2003) 851-868
4. Arandjelovic, O., and Cipolla, R.: Face Recognition from Image Sets using Robust Kernel Resistor-Average Distance. The First IEEE Workshop on Face Processing in Video, (2004)
5. Fukui, K., and Yamaguchi, O.: Face Recognition Using Multi-viewpoint Patterns for Robot Vision. 11th International Symposium of Robotics Research, (2003) 192-201
6. Sato, T., Sukegawa, H., Yokoi, K., Dobashi, H., Ogata, J., and Okazaki, A.,: "FacePass" – Development of a Face-Recognition Security System Unaffected by Entrant's Stance. The Institute of Image Information and Television Engineers Transactions, Vol.56, No. 7, (2002), 1111-1117, (in Japanese).
7. Kozakaya, T., and Nakai, H.: Development of a Face Recognition System on an Image Processing LSI chip. The First IEEE Workshop on Face Processing in Video, (2004)
8. Breiman, L.: Bagging Predictors. Machine Learning, Vol. 24, No. 2, (1996) 123-140
9. Freund, Y., and Schapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. Journal of Computer and System Sciences, Vol.55, No.1, (1997) 119-139
10. Wang, X., and Tang, X.: Random Sampling LDA for Face Recognition. Proceedings IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, (2004) 259-265
11. Watanabe, S., and Pakvasa, N.: Subspace method of pattern recognition. Proceedings of the 1st International Joint Conference on Pattern Recognition (1973) 25-32
12. Oja, E.: Subspace Methods of Pattern Recognition. Research Studies Press, England, (1983)