Proc. of the 4th Int'l Conf. on Automatic Face and Gesture Recognition(FG2000), pp. 163–168, 2000. Copyrighted by the IEEE.

Comparative Evaluation of Face Sequence Matching for Content-based Video Access

Shin'ichi Satoh
NACSIS
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112-8640, Japan
satoh@rd.nacsis.ac.jp

Abstract

The paper presents comparative evaluation of matching methods of face sequences obtained from actual videos. Face information is quite important in videos, especially in news programs, dramas, and movies. Accurate face sequence matching enables many multimedia applications including content-based face retrieval, automated face annotation, video authoring, etc. However, face sequences in videos are subject to variation in lighting condition, pose, facial expression, etc., which cause difficulty in face matching. In order to cope with this problem, several face sequence matching methods are proposed by extending face still image matching, traditional pattern recognition, and recent pattern recognition techniques. They are expected to be applicable to face sequences extracted from actual videos. The performance of these methods are evaluated as the accuracy of face sequence annotation using the methods. The accuracy is evaluated using considerable amount of actual drama videos. The evaluation results reveal merits and demerits of these methods, and indicate future research direction of face matching for videos.

1. Introduction

Face information is quite important in videos, especially in news programs, dramas, and movies. By extracting face sequences from such videos, multimedia applications having content-based access to face information might become realizable. However, we may at least need to be able to tell which face sequences may correspond to the same person in order to realize multimedia applications such as content-based face retrieval, face annotation, video authoring, etc., which are desired for the next generation video processing.

Face images in videos are subject to variation in lighting condition, pose, facial expression, etc., which cause difficulty in face matching. There have been many research efforts of face matching, including face feature-based methods [3], image-based methods [1, 5, 9], combination of feature- and image-based methods [11], etc. Most of them are, however, concentrated only on still images rather than faces in videos. Moreover, most of them can handle very limited variation, i.e., developed for high-quality experimental images taken under strictly controlled lighting condition with fixed pose and facial expression. Contrary, face sequence matching which achieves good performance for the real video domain is desired.

In this paper, several face sequence matching methods are proposed by extending face still image matching, traditional pattern recognition, and recent pattern recognition techniques. They are expected to be applicable to face sequences extracted from actual videos. The performance of these methods are evaluated as the accuracy of face sequence annotation using the methods. The accuracy is evaluated using considerable amount of actual drama videos. The evaluation results reveal merits and demerits of these methods, and indicate future research direction of face matching for videos.

2. Problem description

2.1. Face sequence extraction

As basic data elements for face sequence matching, face sequences are extracted from given videos. We used an equivalent method described in [8]. In this method, face detection is applied first to every frame within a certain interval of frames. In our experiment, we apply the face detector at intervals of 10 frames. The system uses the neural network-based face detector [7], which detects mostly frontal faces of various sizes and at various locations. The face detector can also detect eyes. To ensure that the faces are frontal and close-up, we use only faces in which eyes are successfully detected.

Once a face is detected, the system extracts the skin color model. We assume that human skin color has the Gaussian distribution in (R, G, B) space. The system calculates the mean and covariance matrix of $[R\ G\ B]^T$ vectors of pixels in the detected face region, and uses them as the skin color model in such a way that a pixel having smaller Mahalanobis distance than a certain threshold is a skin color pixel. This model is used to extract skin candidate pixels in the subsequent frames. Based on skin candidate pixels, skin regions are composed using binary image processing methods. The overlap between each of these regions and each of the face regions of the previous frame is evaluated to decide whether one of the skin candidate regions is the succeeding face region. Face region tracking is continued until no succeeding face region is found, or until a scene change is encountered. The method also can extract multiple face sequences appear simultaneously.

The system outputs face sequences from given videos relating detected faces included in each face sequence. Each of these faces is then normalized into a 64×64 image based on the eye positions. The system also applies lighting condition compensation and histogram equalization [7] to each face image to compensate lighting condition variation. Finally, the system outputs face sequences S_i, composing N detected faces $F_{i,k}$ as 64×64 images, which are equivalently 4096-dimensional vectors, having intensity values for elements, where $k = 1, \ldots, N$. To obtain corresponding face images from a face sequence S_i, we define the face sequence function as follows:

$$FS(S_i) \quad = \quad \{F_{i,k}\}. \tag{1}$$

2.2. Face sequence annotation

As a typical application of face sequence matching, face sequence annotation for drama videos is assumed. Figure 1 shows the process overview of face sequence annotation. The system is first given an episode of a drama series, and by having part of face sequences of the episode manually named, the system is trained so that it can annotate unknown face sequences. Then the system is given other episodes of the drama series, and it associates face sequences of the videos to inferred names.

Assume that we have a set of face sequences: $S_i, i = 1, \cdots, N_s$ extracted from given videos. The problem here is how we can realize identification of face sequences, i.e., naming face sequences by the system. Let $\hat{S}_i$ be manually named face sequences, and n_i be corresponding names, respectively, where $i = 1, \cdots, \hat{N}_s$. Each name (n_i) can be one of N_n distinctive names: $\hat{n_j}, j = 1, \cdots, N_n$. Then named face sequences can compose face classes C_j each of which corresponds to a distinctive name $\hat{n_j}$:

$$C_j \quad = \quad \{\hat{S}_i | n_i = \hat{n_j}\}. \tag{2}$$

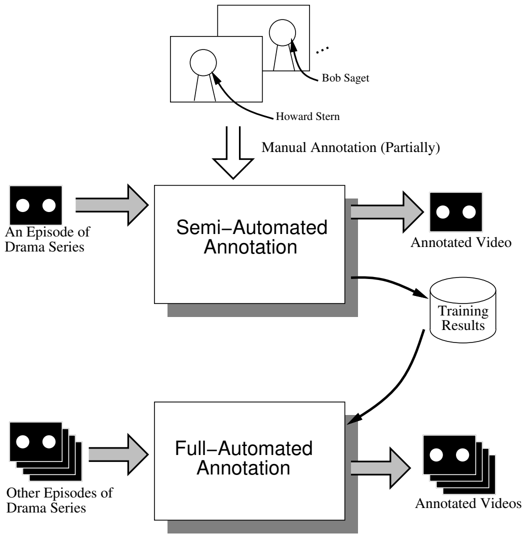

Figure 1. Process of face sequence annotation

Then the problem would be a way how the system associates each face sequence S_i to a particular face class C_j. To obtain corresponding face images from a face class C, we overload the face sequence function as follows:

$$FS(C) \quad = \quad \{F | F \in FS(S) \land S \in C\}. \tag{3}$$

If we somehow define distance between a face sequence and class of face sequence as $d_s(S_i, C_j)$, association between them can be achieved as follows:

$$\text{Associate } S_i \text{ to } C_j \quad \text{if} \quad j = \arg\min_j d_s(S_i, C_j). \tag{4}$$

In the following section, we will propose several face sequence matching methods to calculate face sequence distance $d_s(S_i, C_j)$. The performance of the proposed face sequence matching methods will be evaluated as the accuracy of face sequence annotation. The experimental evaluation reveals merits and demerits for each method.

3. Face sequence matching based on face matching

In order to realize face sequence and face class matching, (face sequence matching for short), we can use face and face matching (face matching for short) as its basis. We can realize face sequence matching of a face sequence and class as face matching of a pair of faces each of which is chosen from the face sequence (class) appropriately. As described before, however, each face image may have very large variation even among face images of the same person, due to

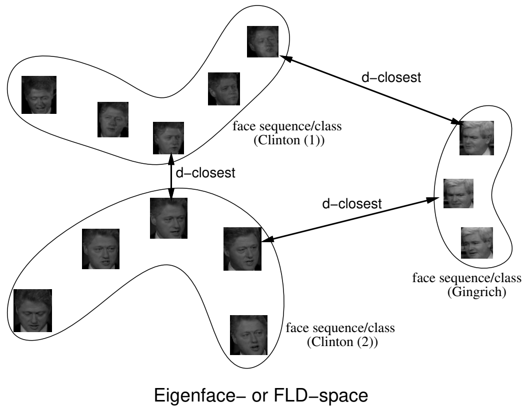

Figure 2. Face sequence distance using face matching.

variation in lighting condition, pose, face expression, etc. It is necessary to choose an appropriate face from a face sequence (class), compensating variation of face images.

To cope with this problem, we can use a pair of faces from a face sequence and face class, which have the same variation. Therefore, we propose face sequence matching based on face matching between a closest pair of faces. This method depends on the presumption that when two face sequences correspond to the same person, the closest pair of them corresponds to faces having the similar pose, facial expression, etc. The face sequence distance is defined as follows:

$$d_s(S,C) \stackrel{\text{def}}{=} \min_{F_s \in FS(S), F_c \in FS(C)} (d_f(F_s, F_c)) \quad (5)$$

where $d_f(F_s, F_c)$ is face distance of face images F_s and F_c. This method is supposed to work fine, because in most cases

- A face sequence surely corresponds to a particular person (no variation by personal identity),
- Each face sequence has sufficient variation in pose, facial expression, etc., not by personal identity, so that there can be a closest pair having the same variation.

Figure 2 depicts the idea of face sequence matching using the closest pair.

3.1. Eigenface-based method

To calculate face distance, the Eigenface-based method [9] can be used. As a training set for the Eigenface, we use all the faces included in named face sequences (a training set). Let X be a training set matrix:

$$X = [x_1\ x_2 \cdots] \quad (6)$$

$$\{x_i\} = \bigcup_k FS(\hat{S}_k). \quad (7)$$

Then the Eigenfaces W^{EIG} can be obtained by using the covariance matrix Σ_X of X and its eigenvalue decomposition:

$$\Sigma_X = \frac{1}{N_s} X X^T \quad (8)$$

$$\Sigma_X W^{EIG} = \Lambda W^{EIG} \quad (9)$$

where W^{EIG} is an eigenvector matrix and Λ is a diagonal eigenvalue matrix. W^{EIG} should include only eigenvectors corresponding to the N^{EIG} largest eigenvalues. By using this, we can define mapping from a face image F to the N^{EIG}-dimensional face vector f:

$$f = {W^{EIG}}^T F. \quad (10)$$

By using this method, we convert each face image to a point in the 16-dimensional Eigenface space, i.e., $N^{EIG} = 16$. Face similarity can then be evaluated as the face distance, i.e., the Euclidean distance between two corresponding points in the Eigenface space:

$$d_f^{EIG}(F_1, F_2) = |f_1 - f_2| \quad (11)$$

$$= |{W^{EIG}}^T F_1 - {W^{EIG}}^T F_2|. \quad (12)$$

Thus we can define face sequence distance $d_s^{EIG}(S,C)$ using the face distance $d_f^{EIG}(F_s, F_c)$:

$$d_s^{EIG}(S,C) = \min_{F_s \in FS(S), F_c \in FS(C)} (d_f^{EIG}(F_s, F_c)). \quad (13)$$

We call this method Eigenface-based face sequence matching.

3.2. Fisher's linear discriminant-based method

The Eigenface-based method transforms each face image into a lower dimensional vector, preserving the mean-square error minimum. In doing this, the method uses only distribution of face images in the image space, but it does not use corresponding names, i.e., class information of faces. We can optimize the linear transformation from face images into points in the lower dimensional space by using the name information.

In order to obtain an optimal linear transformation for classified high-dimensional data, we can use the Fisher's linear discriminant (FLD for short). In [1], FLD was employed to reduce dimensionality for named faces with various lighting condition and fixed pose and facial expression. On the other hand, FLD is applied to faces with various pose and facial expression and with lighting condition compensation in our research.

The linear transform W^{FLD} which transforms an original face image F into the corresponding face vector f, having lower dimensionality N^{FLD}, say, $N^{FLD} = 16$, as follows:

$$f = {W^{FLD}}^T F. \quad (14)$$

The linear transform W^{FLD} can be determined by using FLD with the maximum ratio of between-class scatter to within-class scatter. Between-class scatter S_B and within-class scatter S_W are defined as follows:

$$S_B = \sum_i |FS(C_i)|(M_i - M)(M_i - M)^T \quad (15)$$

$$S_W = \sum_i \sum_{F \in FS(C_i)} (F - M_i)(F - M_i)^T \quad (16)$$

$$M_i = \frac{1}{|FS(C_i)|} \sum FS(C_i) \quad (17)$$

$$M = \frac{1}{\sum_i |FS(C_i)|} \sum_i \sum FS(C_i). \quad (18)$$

An optimal W^{FLD} can be determined by solving an eigenvalue problem:

$$S_B W^{FLD} = \Lambda S_W W^{FLD} \quad (19)$$

where Λ is a diagonal eigenvalue matrix. W^{FLD} should include only eigenvectors corresponding to the N^{FLD} largest eigenvalues. By using W^{FLD}, we can define new face distance:

$$d_f^{FLD}(F_1, F_2) = |f_1 - f_2| \quad (20)$$

$$= |{W^{FLD}}^T F_1 - {W^{FLD}}^T F_2|. \quad (21)$$

Thus we can define face sequence distance $d_s^{FLD}(S, C)$ using the face distance $d_f^{FLD}(F_s, F_c)$:

$$d_s^{FLD}(S, C) = \min_{F_s \in FS(S), F_c \in FS(C)} (d_f^{FLD}(F_s, F_c)). \quad (22)$$

We call this method FLD-based face sequence matching.

4. Face sequence matching based on face class matching

We showed that face matching can realize face sequence matching by choosing the closest pair of a face sequence and class. However, the method could be noisy: for example, there can be a face in a face class which was detected by error and is notoriously far from the others. This face can accidentally be very close to a face sequence which should not be matched to the face class, but will be matched due to the erroneously detected face. In this case, we might eliminate the erroneous face by using the fact that the erroneous face is not statistical majority. Therefore, this problem can be resolved by extracting and using statistical models of face classes.

In the following subsections, we describe methods to obtain statistical models of face classes, and define distance between a face F and face class C using the face class model: $d_c(C, F)$. Then face sequence distance is defined as follows:

$$d_s(S, C) = \min_{F_s \in FS(S)} (d_c(C, F_s)). \quad (23)$$

The basic idea is brought from the closest pair-based face sequence matching, but this might be less sensitive to outliers. This is because the method uses statistical models for face classes, and it is expected that statistically minor outliers will be less effective to the classes.

4.1. Subspace-based method

The CLAFIC method [6] can be used to compose statistical models from face classes as well as calculate the distance between a face and face class. The method first compose the linear subspace for each face class by taking into account the distribution of face images $\{F | F \in FS(C)\} = \{F_1, F_2, \cdots\}$, included in the face class C, as follows:

$$\Sigma_C = E\{F_i {F_i}^T\} \quad (24)$$

$$\Sigma_C W_C^{CLF} = \Lambda W_C^{CLF} \quad (25)$$

where W_C^{CLF} is an eigenvector matrix and Λ is a diagonal eigenvalue matrix. The orthonormal bases for the linear subspace for a class C are selected from W_C^{CLF} corresponding to the N_C^{CLF} largest eigenvalues. Assume that $\lambda_1, \lambda_2, \cdots$ are eigenvalues obtained from Eq. (25) in a decreasing order, then N_C^{CLF} is determined as the largest number so that the cumulative normalized eigenvalue does not exceed a certain threshold:

$$\frac{\sum_{i=1}^{N_C^{CLF}} \lambda_i}{\sum_i \lambda_i} \leq \tau \quad (26)$$

where τ is a scalar number. Each face class may have the different dimension of the linear subspace from the others. The performance of the method can be affected by the choice of τ. So this constant should be chosen carefully.

In the original CLAFIC method, an unknown vector x is given, its projection onto each class is calculated and the vector is classified to the class having the longest projection. For example, let us assume that we have N_n classes $C_i, i = 1, \cdots, N_n$, corresponding to the orthonormal basis matrices $W_{C_i}^{CLF}$. Then the vector x is classified to C_k if:

$$k = \arg\max_k {W_{C_k}^{CLF}}^T x. \quad (27)$$

In our method, we need to define distance between a face F and face class C_i by modifying Eq. (27).

$$d_c^{CLF}(C_i, F) = \frac{\sqrt{|F|^2 - ({W_{C_i}^{CLF}}^T F)^2}}{|F|}. \quad (28)$$

In the original CLAFIC method, the length of projection is compared among classes for the fixed unknown vector.

In contrast to this, since the distance between a face and face class will be compared among classes as well as among faces in our method, we normalize the projection by the norm of F. As described before, face sequence distance between a face sequence and face class is defined as follows:

$$d_s^{CLF}(S,C) = \min_{F_s \in FS(S)} (d_c^{CLF}(C,F_s)). \quad (29)$$

4.2. Kernel function subspace-based method

In the subspace-based method, a face model is extracted as a linear subspace from face images of a face class. However, face images of a person are known to compose a nonlinear manifold [2]. Thus it could be much suitable to use nonlinear models for face classes. In order to compose nonlinear models, we can use the subspace method in the nonlinear space using a kernel function [4]. This method uses the similar technique to the support vector machines(SVM) [10] to define mapping to nonlinear space using a kernel function. Assume that x is a d-dimensional and ϕ is a nonlinear mapping function which maps d-dimensional vector into d_ϕ-dimensional vector. The nonlinear mapping function ϕ is defined using a kernel function k:

$$k(x,y) = \phi(x)^T \phi(y). \quad (30)$$

We use the Gaussian function for the kernel function:

$$k(x,y) = exp(-\frac{|x-y|^2}{2\sigma^2}) \quad (31)$$

where σ is a scalar constant. In this case, d_ϕ is infinity.

Using this technique, face image vectors are mapped into the nonlinear space. Then we apply the CLAFIC-based method in the nonlinear space as described in [4]. In calculating the projection of a face F onto a class C, assume that C includes face images $F_i, i = 1, \cdots, N_C$. A matrix of training vectors in the nonlinear space is composed as follows:

$$\Phi = [\phi(F_1)\ \phi(F_2)\ \cdots\ \phi(F_{N_C})]. \quad (32)$$

Let us think its singular value decomposition:

$$\Phi = U \Lambda V^T. \quad (33)$$

Then the projection of a face F onto a class C, $p(F,C)$, can be calculated as follows:

$$\begin{aligned} p(F,C) &= |U^T \phi(F)| \\ &= |(\Phi V \Lambda^{-1})^T \phi(F)| \\ &= |\Lambda^{-1} V^T \Phi^T \phi(F)| \\ &= |\Lambda^{-1} V^T [k(F_1,F)\ k(F_2,F) \cdots]^T|. \end{aligned} \quad (34)$$

Since V can be derived using an eigenvalue decomposition of $\Phi^T\Phi$ of which (i,j) component is $k(F_i,F_j)$, the projection can be calculated, although the nonlinear space has infinite dimension. Then distance between a face F and face class C and face sequence distance are defined as follows:

$$\begin{aligned} d_c^{KFS}(C,F) &= \sqrt{1-(p(F,C))^2} \quad (35) \\ d_s^{KFS}(S,C) &= \min_{F_s \in FS(S)} (d_c^{KFS}(C,F_s)). \quad (36) \end{aligned}$$

For this method, two parameters σ and τ should be tuned to achieve the best performance.

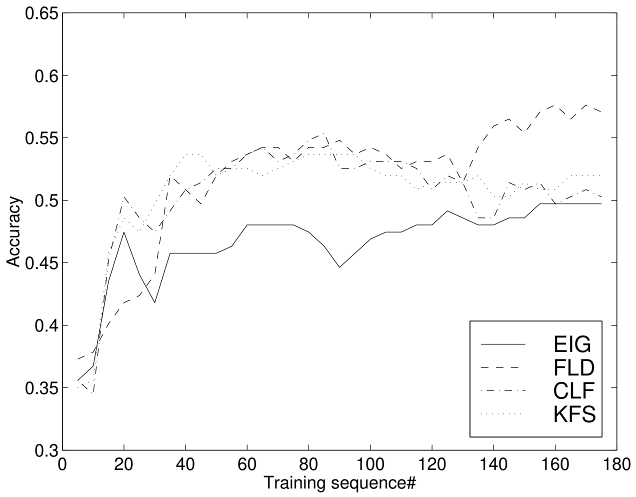

Figure 3. Accuracy of classification

5. Evaluation

Four face sequence matching methods, the Eigenface-based method (EIG), FLD-based method (FLD), subspace-based method (CLF), and kernel function subspace-based method (KFS), are compared by using face sequences obtained from videos. In evaluation, we use actual drama videos, the popular TV drama series in Japan entitled "Oshigoto-desu!" ("It's time to work !"), produced by Fuji Television Network. We use two set of videos, the first episode (1.5 hours) and the second episode (1 hour). 209 and 130 face sequences are obtained from the first and second episode respectively. Part of face sequences of the first episode are manually named and used for the training set. In selecting face sequences to be manually named, face sequences having more faces will be given priority. Then face sequences of the second episode are named automatically according to Eq. (4), and its accuracy is evaluated with varying the number of training face sequences ($\hat{N}_s$).

Figure 3 plots accuracy of classification for each method. According to increase of the number of training face sequences, the accuracy also increases. It is clearly shown that the accuracy for EIG is always smaller than FLD, thus the FLD-based method is superior to the Eigenface-based method. The parameters for CLF (τ) and for KFS (τ and

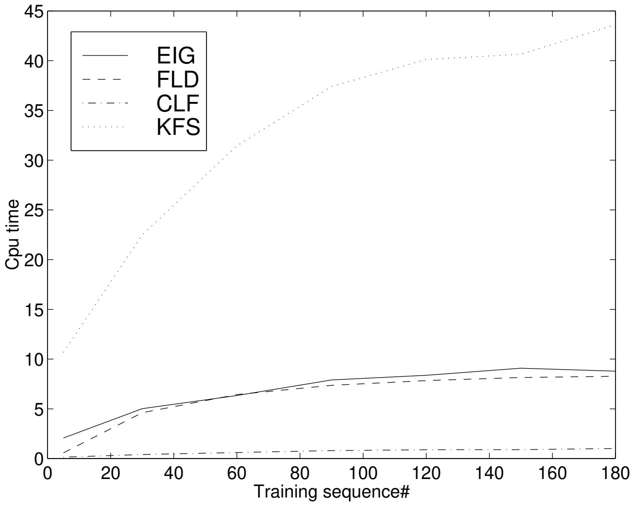

Figure 4. Elapsed time for classification

σ) are optimized manually for $\hat{N}_s = 40$. The graph shows that the accuracy for CLF and KFS (especially KFS) is better than the accuracy for FLD when $\hat{N}_s = 40$. This implies that KFS may beat FLD. However, both CLF and KFS result in much worse accuracy than FLD when $\hat{N}_s$ is different from 40, e.g., $\hat{N}_s$ is around 180. This is due to the parameters for CLF and KFS are sensitive to distribution of a training set pattern. In contrast to this, the FLD-based method achieves the best performance in most cases without requiring parameter tuning.

Due to the nonlinear nature of face manifolds, FLD and KFS are expected to achieve good performance since FLD (and EIG) takes a non-parametric approach and KFS takes an approach in between parametric and non-parametric. It is rather surprising that CLF achieves the comparative performance although CLF uses linear parametric models.

As for classification efficiency, CLF is the fastest because it requires evaluation of Eq. (28) only for each class and for each face included in a query face sequence S_q, i.e., $N_n \times |FS(S_q)|$. On the other hand, EIG and FLD require evaluation of Eq. (12) or (21) for each face in each class and for each face in a query, i.e., $\sum_i^{N_n} |FS(\hat{S}_i)| \times |FS(S_q)|$. KFS requires the same number of evaluation of Eq. (34) as CLF, however, in evaluating this equation, $|FS(\hat{S}_i)|$-times summation is required due to $[k(F_1, F)\ k(F_2, F) \cdots]$ term. Thus the overall order of computation is similar to EIG and FLD. Figure 4 shows elapsed cpu time for each method with varying $\hat{N}_s$. The graph supports the above conjecture.

6. Conclusions

In this paper, face sequence matching methods are explored which can be used for actual videos having heavy variation in lighting condition, pose, face expression, etc. Four of these methods, the Eigenface-based, FLD-based, subspace-based, and kernel function subspace-based methods, are proposed and compared in a real application, the semi-automated face sequence annotation. Experiments reveal that the FLD-based method achieves good performance without requiring parameter tuning, while the kernel function subspace-based method could achieve the best performance only when its parameters are properly tuned. A possible research direction could be pursuing adaptive parameter adjustment. The subspace-based method is the most efficient while preserving comparative classification performance.

Generally, classification accuracy is not sufficiently high for these four methods. A classification error occurs when the system classifies a face sequence to the other class than the correct class, or the correct class is not included in the training set. The latter situation should be treated as "unknown," and a way to discriminate this situation is desired to be achieved.

References

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on PAMI*, 19(7):711–720, 1997.

[2] M. Bichsel and A. P. Pentland. Human face recognition and the face image set's topology. *CVGIP:Image Understanding*, 59(2):254–261, 1994.

[3] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. on PAMI*, 15(10):1042–1052, 1993.

[4] E. Maeda and H. Murase. Kernel based nonlinear subspace method for multi-category classification. In *Proc. of ICASSP*, 1999.

[5] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on PAMI*, 19(7):696–710, 1997.

[6] E. Oja. *Subspace Methods of Pattern Recognition*. Research Studies Press, 1983.

[7] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on PAMI*, 20(1):23–38, 1998.

[8] S. Satoh, Y. Nakamura, and T. Kanade. Name-It: Naming and detecting faces in news videos. *IEEE MultiMedia*, 6(1):22–35, January-March (Spring) 1999.

[9] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[10] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.

[11] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. on PAMI*, 19(7):775–779, 1997.