13

Linear Subspace Learning for Facial Expression Analysis

Caifeng Shan
Philips Research
The Netherlands

1. Introduction

Facial expression, resulting from movements of the facial muscles, is one of the most powerful, natural, and immediate means for human beings to communicate their emotions and intentions. Some examples of facial expressions are shown in Fig. 1. Darwin (1872) was the first to describe in detail the specific facial expressions associated with emotions in animals and humans; he argued that all mammals show emotions reliably in their faces. Psychological studies (Mehrabian, 1968; Ambady & Rosenthal, 1992) indicate that facial expressions, with other non-verbal cues, play a major and fundamental role in face-to-face communication.

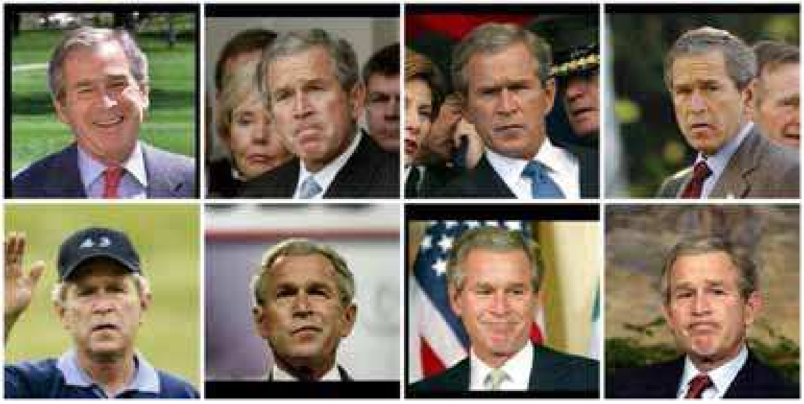


Fig. 1. Facial expressions of George W. Bush.

Machine analysis of facial expressions, enabling computers to analyze and interpret facial expressions as humans do, has many important applications including intelligent human-computer interaction, computer animation, surveillance and security, medical diagnosis, law enforcement, and awareness system (Shan, 2007). Driven by its potential applications and theoretical interests of cognitive and psychological scientists, automatic facial expression analysis has attracted much attention in last two decades (Pantic & Rothkrantz, 2000a; Fasel & Luettin, 2003; Tian et al, 2005; Pantic & Bartlett, 2007). It has been studied in multiple disciplines such as psychology, cognitive science, computer vision, pattern

Source: Machine Learning, Book edited by: Abdelhamid Mellouk and Abdennacer Chebira,
ISBN 978-3-902613-56-1, pp. 450, February 2009, I-Tech, Vienna, Austria

recognition, and human-computer interaction. Although much progress has been made, it is still difficult to design and develop an automated system capable of detecting and interpreting human facial expressions with high accuracy, due to their subtlety, complexity and variability.

Many machine learning techniques have been introduced for facial expression analysis, such as Neural Networks (Tian et al, 2001), Bayesian Networks (Cohen et al, 2003b), and Support Vector Machines (SVM) (Bartlett et al, 2005), to name just a few. Meanwhile, appearance-based statistical subspace learning has been shown to be an effective approach to modeling facial expression space for classification. This is because that despite a facial image space being commonly of a very high dimension, the underlying facial expression space is usually a sub-manifold of much lower dimensionality embedded in the ambient space. Subspace learning is a natural approach to resolve this problem. Traditionally, linear subspace methods including Principal Component Analysis (PCA) (Turk & Pentland, 1991), Linear Discriminant Analysis (LDA) (Belhumeur et al, 1997), and Independent Component Analysis (ICA) (Bartlett et al, 2002) have been used to discover both facial identity and expression manifold structures. For example, Lyons et al (1999) adopted PCA based LDA with the Gabor wavelet representation to classify facial images, and Donato et al (1999) explored PCA, LDA, and ICA for facial action classification.

Recently a number of nonlinear techniques have been proposed to learn the structure of a manifold, e.g., Isomap (Tenenbaum et al, 2000), Local Linear Embedding (LLE) (Roweis & Saul, 2000; Saul & Roweis, 2003), and Laplacian Eigenmaps (Belkin & Niyogi, 2001, 2003). These methods have been shown to be effective in discovering the underlying manifold. However, they are unsupervised in nature and fail to discover the discriminant structure in the data. Moreover, these techniques yield maps that are defined only on the training data, and it is unclear how to evaluate the maps for new test data. So they may not be suitable for pattern recognition tasks such as facial expression recognition. To address this problem, some linear approximations to these nonlinear manifold learning methods have been proposed to provide an explicit mapping from the input space to the reduced space (He & Niyogi, 2003; Kokiopoulou & Saad, 2005). He and Niyogi (2003) developed a linear subspace technique, known as Locality Preserving Projections (LPP), which builds a graph model that reflects the intrinsic geometric structure of the given data space, and finds a projection that respects this graph structure. LPP can be regarded as a linear approximation to Laplacian Eigenmaps; it can easily map any new data to the reduced space by using a transformation matrix. By incorporating the *priori* class information into LPP, we presented a Supervised LPP (SLPP) approach to enhance discriminant analysis on a manifold structure (Shan et al, 2005a). Cai et al (2006) further introduced a Orthogonal LPP (OLPP) approach to produce orthogonal basis vectors, which potentially have more discriminating power.

Orthogonal Neighborhood Preserving Projections (ONPP) is another interesting linear subspace technique proposed recently (Kokiopoulou & Saad, 2005, 2007). ONPP aims to preserve the intrinsic geometry of the local neighborhoods; it can be regarded as a linear approximation to LLE. ONPP constructs a weighted *k*-nearest neighbor graph which models explicitly the data topology, and, similarly to LLE, the weights are decided in a data-driven fashion to capture the geometry of local neighborhoods. In contrast to LLE, ONPP computes an explicit linear mapping from the input space to the reduced space. ONPP can be performed in either an unsupervised or a supervised setting. More recently Cai et al (2007) introduced a linear subspace method called Locality Sensitive Discriminant Analysis

(LSDA), which finds a projection that maximizes the margin between data points from different classes at each local area. LSDA constructs a nearest neighbor graph to model the geometrical structure of the underlying manifold, and then split it into *within-class* graph and *between-class* graph by using the class labels. LPP, ONPP, LSDA are all linear subspace learning techniques which aim at preserving locality of data samples, relying on a nearest neighbor graph to capture the data topology. However, they adopt totally different objective functions, so potentially they will provide different subspace learning power.

As different linear subspace techniques have been developed, the researchers are therefore confronted with a choice of algorithms with significantly different strengthes. However, to our best knowledge, there is no comprehensive comparative study on these linear subspace methods using the same data and experimental settings, although they were individually evaluated. In particular, for the task of facial expression analysis, it is necessary and important to identify the most effective linear subspace technique for facial expression representation and classification. In this chapter, we investigate and evaluate a number of linear subspace techniques for modeling facial expression subspace. Specifically we compare LPP and its variants SLPP and OLPP, ONPP, LSDA with the traditional PCA and LDA using different facial representations on several public databases. We find in our extensive study that the supervised LPP provides the best results in learning facial expression subspace, resulting in superior facial expression recognition performance. A short version of our work was presented in (Shan et al, 2006a).

The remainder of this chapter is organized as follows. We first survey the state of the art of facial expression analysis with machine learning (Section 2). Different linear subspace techniques compared in this chapter are described in Section 3. We present extensive experiments on different databases in Section 4, and finally Section 5 concludes the chapter.

2. State of the art

After Suwa et al (1978) made an early attempt to automatically analyze facial expressions from image sequences, machine analysis of facial expressions has received much attention in last two decades (Pantic & Rothkrantz, 2000a; Fasel & Luettin, 2003; Tian et al, 2005; Pantic & Bartlett, 2007). In this section, we review the state of the art on applying machine learning techniques for facial expression analysis.

Facial expressions can be described at different levels. Two mainstream description methods are facial affect (emotion) and facial muscle action (action unit) (Pantic & Bartlett, 2007). Most of facial expression analysis systems developed so far target facial affect analysis, attempting to analyze a set of prototypic emotional facial expressions (Pantic & Rothkrantz, 2000a, 2003). To describe subtle facial changes, Facial Action Coding System (FACS) (Ekman et al, 2002) has been used for manually labeling of facial actions. FACS associates facial changes with actions of the muscles that produce them. It defines 44 different action units (AUs). Another possible descriptor is the bipolar dimensions of *Valence* and *Arousal* (Russell, 1994). Valence describes the pleasantness, with positive (pleasant) on one end (e.g. happiness), and negative (unpleasant) on the other (e.g. disgust). The other dimension is arousal or activation, for example, sadness has low arousal, whereas surprise has a high arousal level.

The general approach to automatic facial expression analysis consists of three steps: face acquisition, facial data extraction & representation, and facial expression recognition. In the following sections, we discuss these steps respectively.

2.1 Face acquisition

Face acquisition is a pre-processing stage to automatically detect or locate the face region in the input images or sequences. Numerous techniques have been proposed for face detection (Yang et al, 2002), due to its practical importance in many computer vision applications. Most of existing methods emphasize statistical learning techniques and use appearance features. The real-time face detection scheme proposed by Viola and Jones (2001) is arguably the most commonly employed face detector, which consists of a cascade of classifiers trained by AdaBoost employing Harr-wavelet features. AdaBoost (Freund & Schapire, 1997; Schapire & Singer, 1999) is one of the most successful machine learning techniques applied in computer vision area, which provides a simple yet effective approach for stagewise learning of a nonlinear classification function. AdaBoost learns a small number of weak classifiers whose performance are just better than random guessing, and boosts them iteratively into a strong classifier of higher accuracy. Lienhart et al (2003) later made some extensions to this face detector. Many other machine learning techniques such as Neural Networks and SVM have also been introduced for face detection; details can be found in (Yang et al, 2002).

Most of face detectors can only detect faces in frontal or near-frontal view. To handle large head motion in video sequences, head tracking and head pose estimation can be adopted. The tasks of head tracking and pose estimation can be performed sequentially or jointly. Different approaches have been developed for head pose estimation (Murphy-Chutorian & Trivedi, 2008). Given the success of frontal face detectors, a natural extension is to estimate head pose by training multiple face detectors, each to specific a different discrete pose. For example, cascade AdaBoost detectors have been extended for pose estimation (Jones & Viola, 2003). Recently manifold learning approaches have been adopted to seek low-dimensional manifolds that model the continuous variation in head pose; new images can then be embedded into these manifolds for pose estimation. Nonlinear methods such as Isomap, LLE, and Laplacian Eigenmaps or their linear approximations have been exploited for pose estimation (Fu & Huang, 2006; Balasubramanian et al, 2008).

2.2 Facial feature extraction & representation

Facial feature extraction and representation is to derive a set of features from original face images which are used for representing faces. Two types of features, geometric features and appearance features, are usually considered for facial representation. Geometric features deal with the shape and locations of facial components (including mouth, eyes, brows, and nose), which are extracted to represent the face geometry (Zhang et al, 1998; Pantic & Rothkrantz, 2000b; Tian et al, 2001; Kaliouby & Robinson, 2004; Zhang & Ji, 2005; Pantic & Bartlett, 2007). Appearance features present the appearance changes (skin texture) of the face (including wrinkles, bulges and furrows), which are extracted by applying image filters to either the whole face or specific facial regions (Lyons et al, 1999; Donato et al, 1999; Bartlett et al, 2003; Shan et al, 2005c; Littlewort et al, 2006; Gritti et al, 2008). The geometric features based facial representations commonly require accurate and reliable facial feature detection and tracking, which is difficult to accommodate in real-world unconstrained scenarios, e.g., under head pose variation. In contrast, appearance features suffer less from issues of initialization and tracking errors, and can encode changes in skin texture that are critical for facial expression modeling. However, most of the existing appearance-based facial representations still require face registration based on facial feature detection, e.g., eye detection.

Machine learning techniques have been exploited to select the most effective features for facial representation. Donato et al (1999) compared different techniques to extract facial features, which include PCA, LDA, LDA, Local Feature Analysis, and local principal components. The experimental results provide evidence for the importance of using local filters and statistical independence for facial representation. Bartlett et al (2003, 2005) presented to select a subset of Gabor filters using AdaBoost. Similarly, Wang et al (2004) learned a subst of Harr features using Adaboost. Whitehill and Omlin (2006) compared Gabor filters, Harr-like filters, and the edge-oriented histogram for AU recognition, and found that AdaBoost performs better with Harr-like filters, while SVMs perform better with Gabor filters. Valstar and Pantic (2006) recently presented a fully automatic AU detection system that can recognize AU temporal segments using a subset of most informative spatio-temporal features selected by AdaBoost. In our previous work (Shan et al, 2005b; Shan & Gritti, 2008), we also adopted boost learning to learn discriminative Local Binary Patterns features for facial expression recognition.

2.3 Facial expression recognition

The last stage is to classify different expressions based on the extracted facial features. Facial expression recognition can be generally divided into image-based or sequence-based. The image-based approaches use features extracted from a single image to recognize the expression of that image, while the sequence-based methods aim to capture the temporal pattern in a sequence to recognize the expression for one or more images. Different machine learning techniques have been proposed, such as Neural Network (Zhang et al, 1998; Tian et al, 2001), SVM (Bartlett et al, 2005, 2003), Bayesian Network (Cohen et al, 2003b,a), and rule-based classifiers (Pantic & Rothkrantz, 2000b) for image-based expression recognition, or Hidden Markov Model (HMM) (Cohen et al, 2003b; Yeasin et al, 2004) and Dynamic Bayesian Network (DBN) (Kaliouby & Robinson, 2004; Zhang & Ji, 2005) for sequence-based expression recognition.

Pantic and Rothkrantz (2000b) performed facial expression recognition by comparing the AU-coded description of an observed expression against rule descriptors of six basic emotions. Recently they further adopted the rule-based reasoning to recognize action units and their combination (Pantic & Rothkrantz, 2004). Tian et al (2001) used a three-layer Neural Network with one hidden layer to recognize AUs by a standard back-propagation method. Cohen et al (2003b) adopted Bayesian network classifiers to classify a frame in video sequences to one of the basic facial expressions. They compared Naive-Bayes classifiers where the features are assumed to be either Gaussian or Cauchy distributed, and Gaussian Tree-Augmented Naive Bayes classifiers. Because it is difficult to collect a large amount of training data, Cohen et al (2004) further proposed to use unlabeled data together with labeled data using Bayesian networks. As a powerful discriminative machine learning technique, SVM has been widely adopted for facial expression recognition. Recently Bartlett et al (2005) performed comparison of AdaBoost, SVM, and LDA, and best results were obtained by selecting a subset of Gabor filters using AdaBoost and then training SVM on the outputs of the selected filters. This strategy is also adopted in (Tong et al, 2006; Valstar & Pantic, 2006).

Psychological experiments (Bassili, 1979) suggest that the dynamics of facial expressions are crucial for successful interpretation of facial expressions. HMMs have been exploited to capture temporal behaviors exhibited by facial expressions (Oliver et al, 2000; Cohen et al, 2003b; Yeasin et al, 2004). Cohen et al (2003b) proposed a multi-level HMM classifier, which allows not only to perform expression classification in a video segment, but also to automatically segment an arbitrary long video sequence to the different expressions segments without resorting to heuristic methods of segmentation. DBNs are graphical probabilistic models which encode dependencies among sets of random variables evolving in time. HMM is the simplest kind of DBNs. Zhang and Ji (2005) explored the use of multisensory information fusion technique with DBNs for modeling and understanding the temporal behaviors of facial expressions in image sequences. Kaliouby and Robinson (2004) proposed a system for inferring complex mental states from videos of facial expressions and head gestures in real-time. Their system was built on a multi-level DBN classifier which models complex mental states as a number of interacting facial and head displays. Facial expression dynamics can also be captured in low dimensional manifolds embedded in the input image space. Chang et al (2003, 2004) made attempts to learn the structure of the expression manifold. In our previous work (Shan et al, 2005a, 2006b), we presented to model facial expression dynamics by discovering the underlying low-dimensional manifold.

3. Linear subspace methods

The goal of subspace learning (or dimensionality reduction) is to map the data set in the high dimensional space to the lower dimensional space such that certain properties are preserved. Examples of properties to be preserved include the global geometry and neighborhood information. Usually the property preserved is quantified by an objective function and the dimensionality reduction problem is formulated as an optimization problem. The generic problem of linear dimensionality reduction is the following. Given a multi-dimensional data set $x_1,x_2, \ldots ,x_m$ in R^n, find a transformation matrix W that maps these m points to $y_1,y_2, \ldots ,y_m$ in $R^l (l \ll n)$, such that y_i represent x_i, where $y_i = W^T x_i$. In this section, we briefly review the existing linear subspace methods PCA, LDA, LPP, ONPP, LSDA, and their variants.

3.1 Principle Component Analysis (PCA)

Two of the most popular techniques for linear subspace learning are PCA and LDA. PCA (Turk & Pentland, 1991) is an eigenvector method designed to model linear variation in high-dimensional data. PCA aims at preserving the global variance by finding a set of mutual orthogonal basis functions that capture the directions of maximum variance in the data.

Let **w** denote a transformation vector, the objective function is as follows:

$$\max_{\mathbf{w}} \sum_i (y_i - \bar{y})^2 \quad \text{where} \quad \bar{y} = \frac{1}{m} \sum_i y_i \tag{1}$$

The solution $\mathbf{w}_0, \ldots ,\mathbf{w}_{l-1}$ is an orthonormal set of vectors representing the eigenvector of the data’s covariance matrix associated with the l largest eigenvalues.

3.2 Linear Discriminant Analysis (LDA)

While PCA is an unsupervised method and seeks directions that are efficient for representation, LDA (Belhumeur et al, 1997) is a supervised approach and seeks directions that are efficient for discrimination. LDA searches for the projection axes on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other.

Suppose the data samples belong to *c* classes, The objective function is as follows:

$$\max_{\mathbf{w}} \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}} \tag{2}$$

$$S_B = \textstyle\sum_{i=1}^{c} n_i(\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T \tag{3}$$

$$S_W = \textstyle\sum_{i=1}^{c} \left(\sum_{j=1}^{n_i} (x_j^{(i)} - \mathbf{m}^{(i)})(x_j^{(i)} - \mathbf{m}^{(i)})^T \right) \tag{4}$$

where **m** is the mean of all the samples, n_i is the number of samples in the *i*th class, $\mathbf{m}_{(i)}$ is the average vector of the *i*th class, and $x_j^{(i)}$ is the *j*th sample in the *i*th class.

3.3 Locality Preserving Projections (LPP)

LPP (He & Niyogi, 2003) seeks to preserve the intrinsic geometry of the data by preserving locality. To derive the optimal projections preserving locality, LPP employs the same objective function with Laplacian Eigenmaps:

$$\min_{\mathbf{w}} \sum_{i,j} \left(\mathbf{w}^T x_i - \mathbf{w}^T x_j\right)^2 S_{ij} \tag{5}$$

where S_{ij} evaluates a local structure of the data space, and is defined as:

$$S_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}} & \text{if } x_i \text{ and } x_j \text{ are "close"} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

or in a simpler form as

$$S_{ij} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are "close"} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where "close" can be defined as $\|x_i - x_j\|^2 < \varepsilon$, where ε is a small constant, or x_i is among ***k*** nearest neighbors of x_j or x_j is among ***k*** nearest neighbors of x_i. The objective function with symmetric weights $S_{ij}(S_{ij} = S_{ji})$ incurs a heavy penalty if neighboring points x_i and x_j are mapped far apart. Minimizing their distance is therefore an attempt to ensure that if x_i and x_j are "close", $y_i(= \mathbf{w}^T x_i)$ and $y_j(= \mathbf{w}^T x_j)$ are also "close". The objective function of Eqn. (5) can be reduced to:

$$\begin{aligned} \frac{1}{2}\sum_{ij}(\mathbf{w}^T x_i - \mathbf{w}^T x_j)^2 S_{ij} &= \sum_i \mathbf{w}^T x_i D_{ii} x_i^T \mathbf{w} - \sum_{ij} \mathbf{w}^T x_i S_{ij} x_j^T \mathbf{w} \\ &= \mathbf{w}^T X(D - S)X^T \mathbf{w} \\ &= \mathbf{w}^T XLX^T \mathbf{w} \end{aligned} \tag{8}$$

where $X = [x_1,x_2, \dots ,x_m]$ and D is a diagonal matrix whose entries are column (or row, since S is symmetric) sums of S, $D_{ii} = \sum_j S_{ji}$. $L = D–S$ is a Laplacian matrix. D measures the local density on the data points. The bigger the value D_{ii} is (corresponding to y_i), the more important is y_i. Therefore, a constraint is imposed as follows:

$$\mathbf{y}^T D\mathbf{y} = 1 \Rightarrow \mathbf{w}^T XDX^T \mathbf{w} = 1 \tag{9}$$

The transformation vector $\mathbf{w}$ that minimizes the objective function is given by the minimum eigenvalue solution to the following generalized eigenvalue problem:

$$XLX^T \mathbf{w} = \lambda XDX^T \mathbf{w} \tag{10}$$

Suppose a set of vectors $\mathbf{w}_0, \dots ,\mathbf{w}_{l-1}$ is the solution, ordered according to their eigenvalues, $\lambda_0, \dots ,\lambda_{l-1}$, the transformation matrix is derived as $W =[\mathbf{w}_0,\mathbf{w}_1, \dots ,\mathbf{w}_{l-1}]$.

3.3.1 Supervised Locality Preserving Projections (SLPP)

When the class information is available, LPP can be performed in a supervised manner. We introduced a Supervised LPP to encode more discriminative power than the original LPP for improving classification capacity (Shan et al, 2005a). SLPP preserves the class information when constructing a neighborhood graph such that the local neighborhood of a sample x_i from class c is composed of samples belonging to class c only. This can be achieved by increasing the distances between samples belonging to different classes, but leaving them unchanged if they are from the same class. Let $Dis(i, j)$ denote the distance between x_i and x_j, the distance after incorporating the class information is then

$$SupDis(i, j) = Dis(i, j) + M\delta(i, j) \tag{11}$$

where $M = \max_{i,j} Dis(i, j)$, and $\delta(i, j) = 1$ if x_i and x_j belong to different classes, and 0 otherwise. In this way, distances between samples in different classes will be larger than the maximum distance in the entire data set, so neighbors of a sample will always be picked from the same class. SLPP preserves both local structure and class information in the embedding, so that it better describes the intrinsic structure of a data space containing multiple classes.

3.3.2 Orthogonal Locality Preserving Projections (OLPP)

The basis vectors derived by LPP can be regarded as the eigenvectors of the matrix $(XDX^T)^{-1}XLX^T$ corresponding to the smallest eigenvalues. Since $(XDX^T)^{-1}XLX^T$ is not symmetric in general, these basis vectors are non-orthogonal. Cai et al (2006) presented Orthogonal LPP to enforce the mapping to be orthogonal. The orthogonal basis vectors $\{\mathbf{w}_0,\mathbf{w}_1, \dots ,\mathbf{w}_{l-1}\}$ are computed as follows.

- Compute $\mathbf{w}_0$ as the eigenvector of $(XDX^T)^{-1}XLX^T$ associated with the smallest eigenvalue.
- Compute $\mathbf{w}_k$ as the eigenvector of

$$M^{(k)} = \left(I - (XDX^T)^{-1}A^{(k-1)}(B^{(k-1)})^{-1}(A^{(k-1)})^T\right)(XDX^T)^{-1}XLX^T \tag{12}$$

associated with the smallest eigenvalue, where

$$A^{(k-1)} = [\mathbf{w}_0, \ldots, \mathbf{w}_{k-1}] \tag{13}$$

$$B^{(k-1)} = (A^{(k-1)})^T (XDX^T)^{-1} A^{(k-1)} \tag{14}$$

OLPP can be applied under supervised or unsupervised mode. In this chapter, for facial expression analysis, OLPP is performed in the supervised manner as described in Section 3.3.1.

3.4 Orthogonal Neighborhood Preserving Projections (ONPP)

ONPP (Kokiopoulou & Saad, 2005, 2007) seeks an orthogonal mapping of a given data set so as to best preserve the local geometry. The first step of ONPP, identical with that of LLE, builds an affinity matrix by computing optimal weights which reconstruct each sample by a linear combination of its *k* nearest neighbors. The reconstruction errors are measured by minimizing

$$\varepsilon(V) = \sum_i |x_i - \sum_j v_{ij} x_j|^2 \tag{15}$$

The weights $v_{i\,j}$ represent the linear coefficient for reconstructing the sample x_i from its neighbors $\{x_{\,j}\}$. The following constraints are imposed on the weights:

1. $v_{i\,j} = 0$, if x_j is not one of the *k* nearest neighbors of x_i.
2. $\sum_j v_{i\,j} = 1$, that is x_i is approximated by a convex combination of its neighbors.

In the second step, ONPP derives an explicit linear mapping from the input space to the reduced space. ONPP imposes a constraint that each data sample y_i in the reduced space is reconstructed from its *k* nearest neighbors by the same weights used in the input space, so it employs the same objective function with LLE:

$$\min_Y \sum_i |y_i - \sum_j v_{ij} y_j|^2 \tag{16}$$

where the weights $v_{i\,j}$ are fixed. The optimization problem can be reduced to

$$\begin{aligned} \min_Y \sum_i |y_i - \sum_j v_{ij} y_j|^2 &= \min_W \sum_i |W^T x_i - \sum_j v_{ij} W^T x_j|^2 \\ &= \min_W \|W^T X (I - V^T)\|_F^2 \\ &= \min_W tr(W^T X (I - V^T)(I - V) X^T W) \\ &= \min_W tr(W^T X M X^T W) \end{aligned} \tag{17}$$

where $M = (I–V^T)(I–V)$. By imposing an additional constraint that the columns of W are orthogonal, the solution to the above optimization problem is the eigenvectors associated with the *d* smallest eigenvalues of the matrix

$$\tilde{M} = X(I - V^T)(I - V)X^T. \tag{18}$$

ONPP can be performed in either an unsupervised or a supervised setting. In the supervised ONPP, when building the data graph, an edge exists between x_i and x_j if and only if x_i and x_j belong to the same class. This means that the adjacent data samples in the nearest neighbor graph belong to the same class. So there is no need to set parameter k in the supervised ONPP.

3.5 Locality Sensitive Discriminant Analysis (LSDA)

Given a data set, LSDA (Cai et al, 2007) constructs two graphs, *within-class graph* G_w and *between-class graph* G_b, in order to discover both geometrical and discriminant structure of the data. For each data sample x_i, let $N(x_i)$ be the set of its k nearest neighbors. $N(x_i)$ can be naturally split into two subsets, $N_b(x_i)$ and $N_w(x_i)$. $N_w(x_i)$ contains the neighbors sharing the same label with x_i, while $N_b(x_i)$ contains neighbors have different labels. Let S_w and S_b be the weight matrices of G_w and G_b respectively, which can be defined as follows

$$S_{b,ij} = \begin{cases} 1 & \text{if } x_i \in N_b(x_j) \text{ or } x_j \in N_b(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$S_{w,ij} = \begin{cases} 1 & \text{if } x_i \in N_w(x_j) \text{ or } x_j \in N_w(x_i) \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

To derive the optimal projections, LSDA optimizes the following objective functions

$$\min_{\mathbf{w}} \sum_{i,j} \left(\mathbf{w}^T x_i - \mathbf{w}^T x_j\right)^2 S_{w,ij} \tag{21}$$

$$\max_{\mathbf{w}} \sum_{i,j} \left(\mathbf{w}^T x_i - \mathbf{w}^T x_j\right)^2 S_{b,ij} \tag{22}$$

Similar to Eqn (8), the objective function (21) can be reduced to

$$\begin{aligned} \frac{1}{2}\sum_{ij}(\mathbf{w}^T x_i - \mathbf{w}^T x_j)^2 S_{w,ij} &= \sum_i \mathbf{w}^T x_i D_{w,ii} x_i^T \mathbf{w} - \sum_{ij} \mathbf{w}^T x_i S_{w,ij} x_j^T \mathbf{w} \\ &= \mathbf{w}^T X(D_w - S_w)X^T \mathbf{w} \\ &= \mathbf{w}^T X L_w X^T \mathbf{w} \end{aligned} \tag{23}$$

where D_w is a diagonal matrix, and its entries $D_{w,ii} = \sum_j S_{w,ji}$. Similarly, the objective function (22) can be reduced to

$$\frac{1}{2}\sum_{ij}(\mathbf{w}^T x_i - \mathbf{w}^T x_j)^2 S_{b,ij} = \mathbf{w}^T X(D_b - S_b)X^T \mathbf{w} = \mathbf{w}^T X L_b X^T \mathbf{w} \tag{24}$$

Similar to LPP, a constraint is imposed as follows:

$$\mathbf{y}^T D_w \mathbf{y} = 1 \Rightarrow \mathbf{w}^T X D_w X^T \mathbf{w} = 1 \tag{25}$$

The transformation vector **w** that minimizes the objective function is given by the maximum eigenvalue solution to the generalized eigenvalue problem:

$$X(\alpha L_b + (1-\alpha)S_w)X^T \mathbf{w} = \lambda X D_w X^T \mathbf{w} \tag{26}$$

In practice, the dimension of the feature space (n) is often much larger than the number of samples in a training set (m), which brings problems to LDA, LPP, ONPP, and LSDA. To overcome this problem, the data set is first projected into a lower dimensional space using PCA.

4. Experiments

In this section, we evaluate the above linear subspace methods for facial expression analysis with the same data and experimental settings. We use implementations of LPP, SLPP, OLPP, ONPP and LSDA provided by the authors.

Psychophysical studies indicate that basic emotions have corresponding universal facial expressions across all cultures (Ekman & Friesen, 1976). This is reflected by most current facial expression recognition systems that attempt to recognize a set of prototypic emotional expressions including disgust, fear, joy, surprise, sadness and anger (Lyons et al, 1999; Cohen et al, 2003b; Tian, 2004; Bartlett et al, 2005). In this study, we also focus on these prototypic emotional expressions. We conducted experiments on three public databases: the Cohn-Kanade Facial Expression Database (Kanade et al, 2000), the MMI Facial Expression Database (Pantic et al, 2005), and the JAFFE Database (Lyons et al, 1999), which are the most commonly used databases in the current facial-expression-research community.

In all experiments, we normalized the original face images to a fixed distance between the two eyes. Facial images of 110×150 pixels, with 256 gray levels per pixel, were cropped from original frames based on the two eyes location. No further alignment of facial features such as alignment of mouth (Zhang et al, 1998), or removal of illumination changes (Tian, 2004) was performed in our experiments. Fig. 2 shows an example of the original image and the cropped face image.

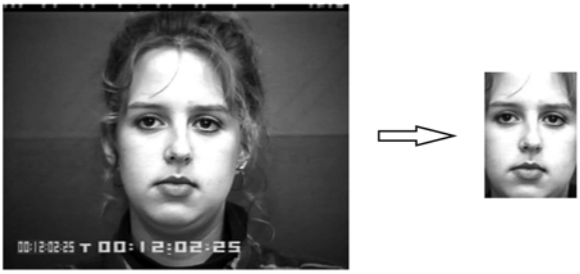


Fig. 2. The original face image and the cropped image.

4.1 Facial representation

To perform facial expression analysis, it is necessary to derive an effective facial representation from original face images. Gabor-wavelet representations have been widely adopted to describe appearance changes of faces (Tian, 2004; Bartlett et al, 2005). However, the computation of Gabor features is both time and memory intensive. In our previous work (Shan et al, 2005c), we proposed Local Binary Patterns (LBP) features as low-cost discriminant appearance features for facial expression analysis. The LBP operator, originally introduced by Ojala et al (2002) for texture analysis, labels the pixels of an image by thresholding a neighborhood of each pixel with the center value and considering the results as a binary number. The histogram of the labels computed over a region can be used as a texture descriptor. The most important properties of the LBP operator are its tolerance against illumination changes and its computational simplicity. LBP features recently have been exploited for face detection and recognition (Ahonen et al, 2004).

In the existing work (Ahonen et al, 2004; Shan et al, 2005c), the face image is equally divided into small regions from which LBP histograms are extracted and concatenated into a single feature histogram (as shown in Fig. 3). However, this LBP feature extraction scheme suffers from fixed LBP feature size and positions. By shifting and scaling a sub-window over face images, many more LBP histograms could be obtained, which yields a more complete description of face images. To minimize the large number of LBP histograms necessarily introduced by shifting and scaling a sub-window, we proposed to learn the most effective LBP histograms using AdaBoost (Shan et al, 2005b). The boosted LBP features provides a compact and discriminant facial representation. Fig. 4 shows examples of the selected subregions (LBP histograms) for each basic emotional expression. It is observed that the selected sub-regions have variable sizes and positions.

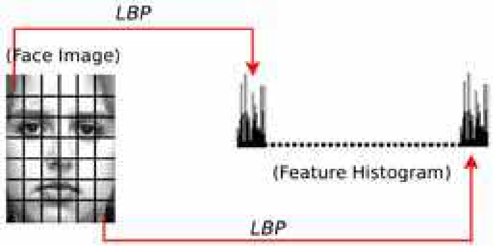


Fig. 3. A face image is divided into small regions from which LBP histograms are extracted and concatenated into a single, spatially enhanced feature histogram.

In this study, three facial representations were considered: raw gray-scale image (IMG), LBP features extracted from equally divided sub-regions (LBP), and Boosted LBP features (BLBP). On IMG features, for computational efficiency, we down-sampled the face images to 55×75 pixels, and represented each image as a 4,125(55×75)-dimensional vector. For LBP features, as shown in Fig. 3, we divided facial images into 42 sub-regions; the 59-bin $LBP_{8,2}^{u2}$ operator (Ojala et al, 2002) was applied to each sub-region. So each image was represented by a LBP histogram with length of 2,478(59×42). For BLBP features, by shifting and scaling a sub-window, 16,640 sub-regions, i.e., 16,640 LBP histograms, were extracted from each face image; AdaBoost was then used to select the most discriminative LBP histograms. AdaBoost training continued until the classifier output distribution for the positive and negative samples were completely separated.

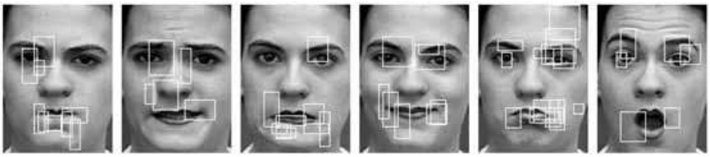


Fig. 4. Examples of the selected sub-regions (LBP histograms) for each of the six basic emotions in the Cohn-Kanade Database (from left to right: Anger, Disgust, Fear, Joy, Sadness, and Surprise).

4.2 Cohn-Kanade database

The Cohn-Kanade Database (Kanade et al, 2000) consists of 100 university students in age from 18 to 30 years, of which 65% were female, 15% were African-American, and 3% were Asian or Latino. Subjects were instructed to perform a series of 23 facial displays, six of which were prototypic emotions. Image sequences from neutral face to target display were digitized into 640×490 pixel arrays. Fig. 5 shows some sample images from the database.

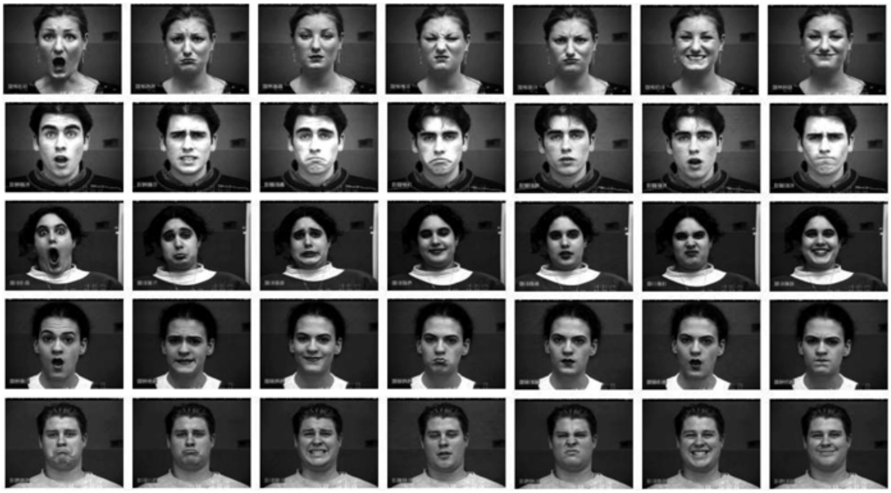


Fig. 5. The sample face expression images from the Cohn-Kanade Database.

In our experiments, 320 image sequences were selected from the database. The only selection criterion is that a sequence can be labeled as one of the six basic emotions. The sequences come from 96 subjects, with 1 to 6 emotions per subject. Two data sets were constructed: (1) **S1**: the three peak frames (typical expression at apex) of each sequence were used for 6-class expression analysis, resulting in 960 images (108 Anger, 120 Disgust, 99 Fear, 282 Joy, 126 Sadness, and 225 Surprise); (2) **S2**: the neutral face of each sequence was further included for 7-class expression analysis, resulting in 1,280 images (960 emotional images plus 320 neutral faces).

4.2.1 Comparative evaluation on subspace learning

As presented in (Shan et al, 2006a), we observed in our experiments on all databases that ONPP and the supervised ONPP achieve comparable performance in expression subspace learning and expression recognition. It seems that the label information used in the supervised ONPP does not provide it with more discriminative power than ONPP for facial expression analysis. Therefore, in this chapter, we focus on the evaluation of the supervised ONPP. We also found in our experiments that the supervised OLPP provides similar results with SLPP, so we mainly focus on the evaluation of SLPP in this chapter.

The 2D visualization of embedded subspaces of data set **S1** is shown in Fig. 6. In the six methods compared, PCA and LPP are unsupervised techniques, while LDA, SLPP, ONPP, and LSDA perform in a supervised manner. It is evident that the classes of different expressions are heavily overlapped in 2D subspaces generated by unsupervised methods PCA and LPP (with all three facial representations), therefore are poorly represented. The

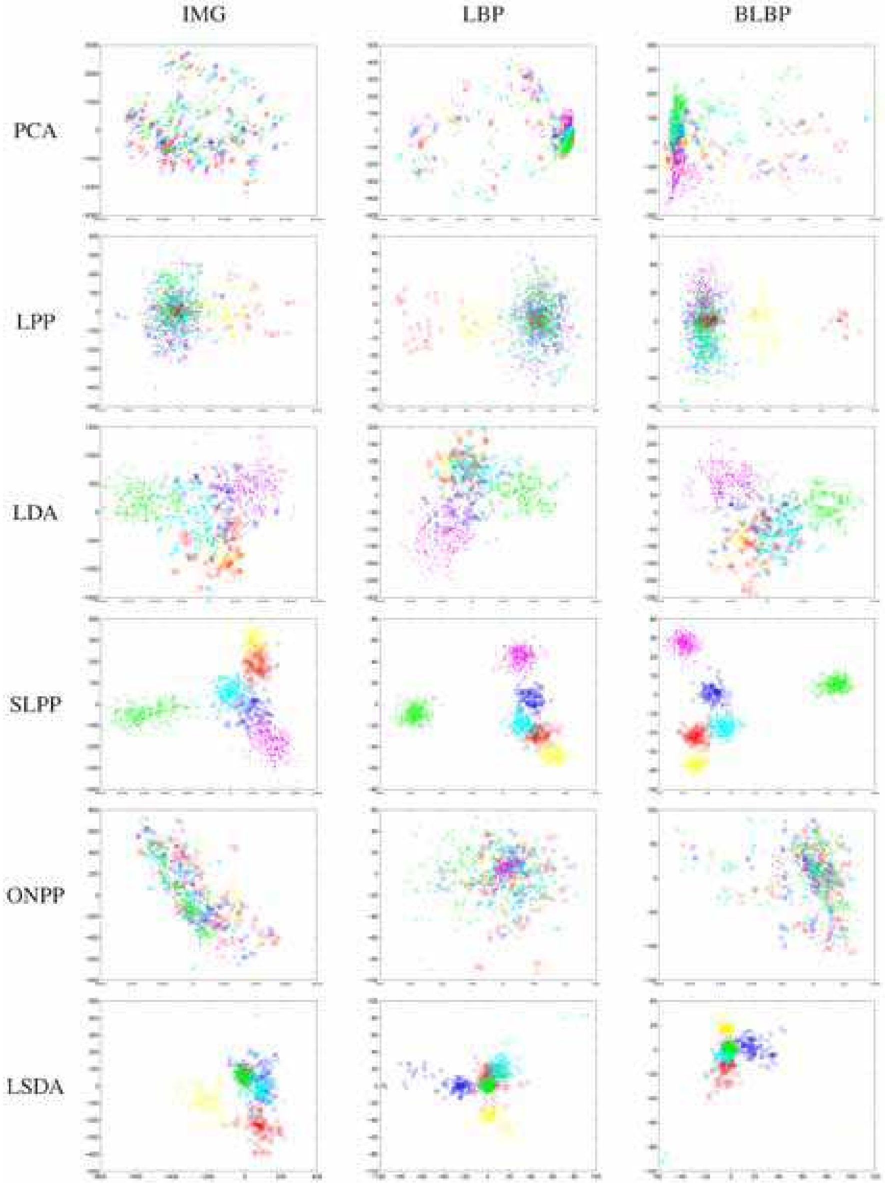


Fig. 6. (Best viewed in color) Images of data set **S1** are mapped into 2D embedding spaces. Different expressions are color coded as: Anger (red), Disgust (yellow), Fear (blue), Joy (magenta), Sadness (cyan), and Surprise (green).

projections of PCA are spread out since PCA aims at maximizing the variance. In the cases of LPP, although it preserves local neighborhood information, as expression images contain complex variations and significant overlapping among different classes, it is difficult for

LPP to yield meaningful projections in the absence of class information. For supervised methods, it is surprising to observe that different expressions are still heavily overlapped in the 2D subspace derived by ONPP. In contrast, the supervised methods LDA, SLPP and LSDA yield much meaningful projections since images of the same class are mapped close to each other. SLPP provides evidently best projections since different classes are well separated and the clusters appear cohesive. This is because SLPP preserves the locality and class information simultaneously in the projections. On the other hand, LDA discovers only the Euclidean structure therefore fails to capture accurately any underlying nonlinear manifold that expression images lie on, resulting in its discriminating power being limited. LSDA obtains better projections than LDA as the clusters of different expressions are more cohesive. On comparing facial representation, BLBP provides evidently the best performance with projected classes more cohesive and clearly separable in the SLPP subspace, while IMG is worst.

Fig. 7 shows the embedded OLPP subspace of data set **S1**.We can see that OLPP provides much similar projections to SLPP. The results obtained by SLPP and OLPP reflect human observation that Joy and Surprise can be clearly separated, but Anger, Disgust, Fear and Sadness are easily confused. This reenforces the findings in other published work (Tian, 2004; Cohen et al, 2003a).

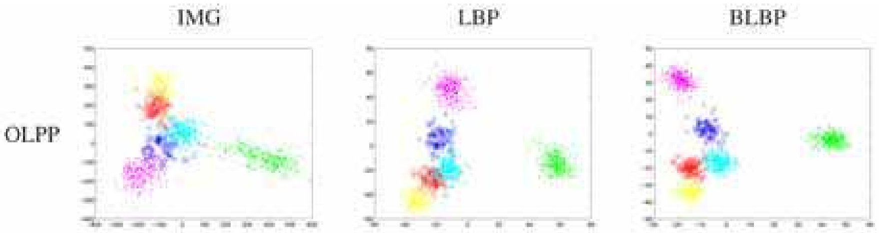
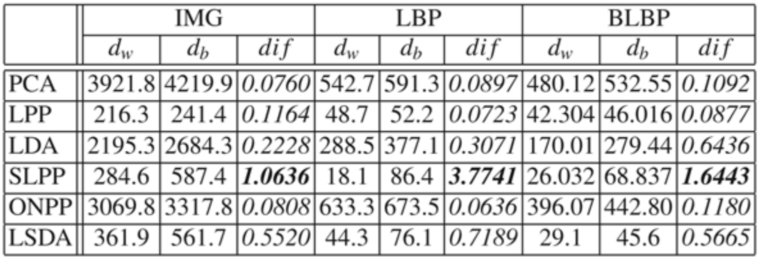


Fig. 7. (Best viewed in color) Images of data set **S1** are mapped into 2D embedding spaces of OLPP.

For a quantitative evaluation of the derived subspaces, following the methodology in (Li et al, 2003), we investigate the histogram distribution of within-class pattern distance and between-class pattern distance of different techniques. The former is the distance between expression patterns of the same expression class, while the latter is the distance between expression patterns belonging to different expression classes. Obviously, for a good representation, the within-class distance distribution should be dense, close to the origin, having a high peak value, and well-separated from the between-class distance distribution.We plot in Fig. 8 the results of different methods on **S1**. It is observed that SLPP consistently provides the best distributions for different facial representations, while those of PCA, LPP, and ONPP are worst. The average within-class distance d_w and between-class distance d_b are shown in Table 1. To ensure the distance measures from different methods are comparable, we compute a normalized difference between the within- and between-class distances of each method as $dif = \frac{d_b - d_w}{d_w}$, which can be regarded as a relative measure on how widely the within-class patterns are separated from the between-class patterns. A high value of this measure indicates success. It is evident in Table 1 that SLPP has the best separating power whilst PCA, LPP and ONPP are the poorest. The separating power of

LDA and LSDA is inferior to that of SLPP, but always outperform those of PCA, LPP, and ONPP. Both Fig. 8 and Table 1 reinforce the observation in Fig. 6.



	IMG			LBP			BLBP		
	d_w	d_b	dif	d_w	d_b	dif	d_w	d_b	dif
PCA	3921.8	4219.9	*0.0760*	542.7	591.3	*0.0897*	480.12	532.55	*0.1092*
LPP	216.3	241.4	*0.1164*	48.7	52.2	*0.0723*	42.304	46.016	*0.0877*
LDA	2195.3	2684.3	*0.2228*	288.5	377.1	*0.3071*	170.01	279.44	*0.6436*
SLPP	284.6	587.4	***1.0636***	18.1	86.4	***3.7741***	26.032	68.837	***1.6443***
ONPP	3069.8	3317.8	*0.0808*	633.3	673.5	*0.0636*	396.07	442.80	*0.1180*
LSDA	361.9	561.7	*0.5520*	44.3	76.1	*0.7189*	29.1	45.6	*0.5665*

Table 1. The average within-class and between-class distance and their normalization difference values on data set **S1**.

The 2D visualization of embedded subspaces of data set **S2** with different subspace techniques and facial representations is shown in Fig. 9. We observe similar results to those obtained in 6-class problem. SLPP outperforms the other methods in derive the meaningful projections. Different expressions are heavily overlapped in 2D subspaces generated by PCA, LPP, and ONPP, and the discriminating power of LDA is also limited.We further show in Fig. 10 the embedded OLPP subspace of data set **S2**, and also observe that OLPP provides much similar projections to SLPP. Notice that in the SLPP and OLPP subspaces, after including neutral faces, Anger, Disgust, Fear, Sadness, and Neutral are easily confused, while Joy and Surprise still can be clearly separated.

4.2.2 Comparative evaluation on expression recognition

To further compare these methods, we also performed facial expression recognition in the derived subspaces. We adopted the *k* nearest-neighbor classifier for its simplicity. The Euclidean metric was used as the distance measure. The number of nearest neighbors was set according to the size of the training set. To evaluate the algorithms' generalization ability, we adopted a 10-fold cross-validation test scheme.

That is, we divided the data set randomly into ten groups of roughly equal numbers of subjects, from which the data from nine groups were used for training and the left group was used for testing. The process was repeated ten times for each group in turn to be tested. We reported the average recognition results (with the standard deviation) here.

The recognition performance of subspace learning techniques varies with the dimensionality of subspace (note that the dimension of the reduced LDA subspace is at most c–1, where c is the number of classes). Moreover, the graph-based techniques rely on the parameter *k*, the number of nearest neighbors used when building the graph; how to set the parameter is still an open problem. In our cross-validation experiments, we tested different combinations of the parameter *k* with the subspace dimensionality, and the best performance obtained are shown in Tables 2 and 3. It is observed that the supervised approaches perform robustly better than the unsupervised methods. For unsupervised methods, PCA performs better than LPP, with all three facial representations. For supervised methods, it is evident that SLPP has a clear margin of superiority over LDA (12-38% better), ONPP (25-64% better), and LSDA (6-13% better). Both LSDA and LDA perform better than ONPP, and LSDA outperforms LDA. On comparing the standard deviation of 10-fold cross validation, SLPP

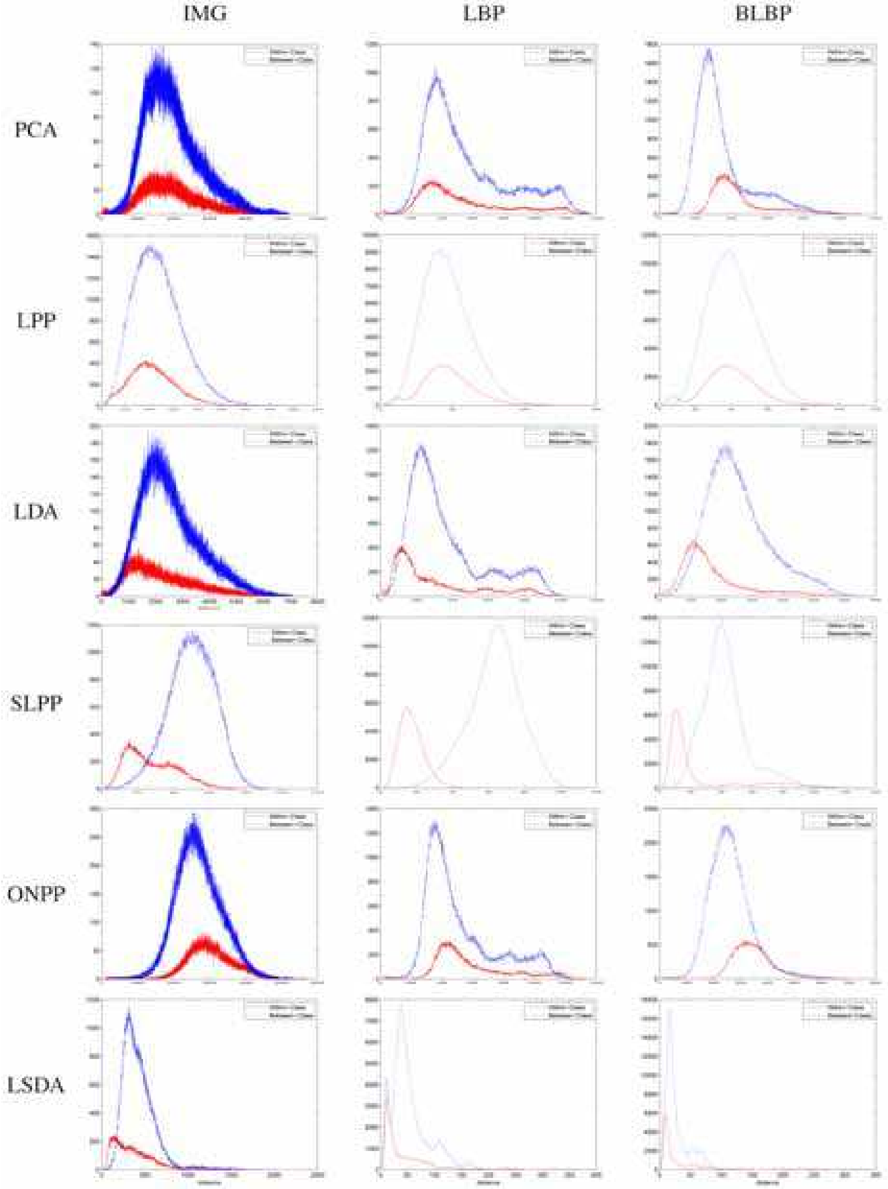


Fig. 8. (Best viewed in color) Histogram distribution of within-class pattern distance (solid red lines) and between-class pattern distances (dotted blue line) on data set **S1**

always produces the smallest deviation (one exception with IMG on **S2**). This demonstrates that SLPP is much more robust than other methods. The recognition results reinforce our early observations shown in Fig. 6, Fig. 8 and Table 1. To clearly compare recognition rates of different methods with different facial representations, we plot the bar graphes of