*Research Article*

# Eigenvector Weighting Function in Face Recognition

## Pang Ying Han,[1] Andrew Teoh Beng Jin,[2,3] and Lim Heng Siong[4]

[1] *Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia*
[2] *School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Republic of Korea*
[3] *Predictive Intelligence Research Cluste, Sunway University, Bandar Sunway, 46150 P. J. Selangor, Malaysia*
[4] *Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia*

Correspondence should be addressed to Andrew Teoh Beng Jin, andrew_tbj@yahoo.com

Graph-based subspace learning is a class of dimensionality reduction technique in face recognition. The technique reveals the local manifold structure of face data that hidden in the image space via a linear projection. However, the real world face data may be too complex to measure due to both external imaging noises and the intra-class variations of the face images. Hence, features which are extracted by the graph-based technique could be noisy. An appropriate weight should be imposed to the data features for better data discrimination. In this paper, a piecewise weighting function, known as Eigenvector Weighting Function (EWF), is proposed and implemented in two graph based subspace learning techniques, namely Locality Preserving Projection and Neighbourhood Preserving Embedding. Specifically, the computed projection subspace of the learning approach is decomposed into three partitions: a subspace due to intra-class variations, an intrinsic face subspace, and a subspace which is attributed to imaging noises. Projected data features are weighted differently in these subspaces to emphasize the intrinsic face subspace while penalizing the other two subspaces. Experiments on FERET and FRGC databases are conducted to show the promising performance of the proposed technique.

## 1. Introduction

In general, a face image with size $m \times n$ can be perceived as a vector in an image space $\mathbf{R}^{m \times n}$. If this high-dimensional vector is input directly for classification, poor performance is expected due to curse of dimensionality [1]. Therefore, an effective dimensionality reduction technique is required to alleviate this problem. Conventionally, the most representative dimensionality reduction techniques include Principal Component Analysis (PCA) [2]

and Linear Discriminant Analysis (LDA) [3]; and they have demonstrated a fairly good performance in face recognition. These algorithms assume the data is Gaussian distributed, but turn out to be not usually assured in practice. Therefore, they may fail to reveal the intrinsic structure of the face data.

Recent studies show the intrinsic geometrical structures of the face data are useful for classification [4]. Hence, a couple of graph-based subspaces learning algorithms has been proposed to reveal the local manifold structure of the face data hidden in the image space [4]. The instances of graph-based algorithms include Locality Preserving Projection (LPP) [5], Locally Linear Discriminate Embedding [6] and Neighbourhood Preserving Embedding (NPE) [7]. These algorithms were shown to unfold the nonlinear structure of the face manifold by means of mapping nearby points in the high-dimensional space to the nearby points in a low-dimensional feature space. They preserve the local neighbourhood relation without imposing any restrictive assumption on the data distribution. In fact, these techniques can be unified with a general framework so-called graph embedding framework with linearization [8]. The dimension reduction problem by means of graph-based subspace learning approach can be boiled down by solving a generalized eigenvalue problem

$$\mathbf{S}_1^{\mathrm{T}} \boldsymbol{\nu} = \beta \mathbf{S}_2 \boldsymbol{\nu}, \tag{1.1}$$

where $\mathbf{S}_1$ and $\mathbf{S}_2$ are the matrices to be minimized and maximized, respectively. Different notions of $\mathbf{S}_1$ and $\mathbf{S}_2$ correspond to different graph-based algorithms. The computed eigenvector, $\boldsymbol{\nu}$ (or eigenspace) will be utilized to project input data into a lower-dimensional feature representation.

There are rooms to further exploit the underlying discriminant property of graph-based subspaces learning algorithms since the real-world face data may be too complex. Face images per subject are varying due to external factors (e.g., sensor noise, unknown noise sources, etc.) and the intraclass variations of the images caused by pose, facial expression and illumination variations. Therefore, features extracted by the subspace learning approach may be noisy and may not be favourable for classification. An appropriate weight should be imposed to the eigenspace for better class discrimination.

In this paper, we propose to decompose the whole eigenspace, constituted by all the eigenvectors computed through (1.1), of subspace learning approach into three subspaces: a subspace due to facial intraclass variations (noise I subspace, **N-I**), an intrinsic face subspace (face subspace, **F**), and a subspace that is attributed to sensor and external noises (noise II subspace, **N-II**). The justification for the eigenspace decomposition will be explained in Section 3. The purpose of the decomposition is to weight the three subspaces differently to stress the informative face dominating eigenvectors, and to demphasize the eigenvectors in the two noise subspaces. Therefore, an effective weighting approach, known as Eigenvector Weighting Function (EWF) is introduced. We apply EWF on LPP and NPE for face recognition.

The main contributions of this work include: (1) the decomposition of the eigenspace of subspace learning approach into noise I, face and noise II subspaces, where the eigenfeatures are weighted differently in these subspaces (2) an effective weighting function that enforces appropriate emphasis or de-emphasis on the eigenspace, and (3) a feature extraction method with an effective eigenvector weighting scheme to extract significant features for data analysis.

The paper is organized as follows: in Section 2, we present a comprehensive description about the Graph Embedding framework, and this is followed by the proposed

Eigenvector Weighting Function (denoted as EWF) in Section 3. We also discuss the numerical justification of EWF in Section 4. The effectiveness of EWF in face recognition is demonstrated in Section 5. Finally, Section 6 contains our conclusion of this study.

## 2. Graph Embedding Framework

In graph embedding framework, each facial image in vector form is represented as a vertex of a graph $G$. Graph embedding transforms the vertex to a low-dimensional vector that preserves the similarities between the vertex pairs [9]. Suppose that we have $n$ numbers of $d$-dimensional face data $\{\mathbf{x}_i \in \mathbf{R}^d \mid i = 1, 2, \ldots, n\}$ and are represented as a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$. Let $G = \{\mathbf{X}, \mathbf{W}\}$ be an undirected weighted graph with vertex set $\mathbf{X}$ and similarity matrix $\mathbf{W} \in \mathbf{R}^{n \times n}$, where $\mathbf{W} = \{W_{ij}\}$ is a symmetric matrix that records the similarity weight of a pair of vertices $i$ and $j$.

Consider that all vertices of the graph are mapped onto a line and $\mathbf{y} = [y_1, y_2, \ldots, y_n]^{\mathrm{T}}$ be such a map. The target is to make the vertices of the graph stay as close as possible. Hence, a graph-preserving criterion is defined as

$$\mathbf{y}^* = \arg\min \sum_{i,j} (y_i - y_j)^2 W_{ij} \tag{2.1}$$

under certain constraints [10]. This objective function ensures that $y_i$ and $y_j$ are close if larger similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. With some simple algebraic tricks, (2.1) can be expressed as

$$\sum_{i,j} (y_i - y_j)^2 W_{ij} = 2\mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{y}, \tag{2.2}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix [9] and $\mathbf{D}$ is a diagonal matrix whose entries are column (or row, since $W$ is symmetric) sums of $\mathbf{W}$, $D_{ii} = \sum_j W_{ji}$. Finally, the minimization problem reduces to,

$$\mathbf{y}^* = \arg\min_{\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}=1} \mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{y} = \arg\min \frac{\mathbf{y}^{\mathrm{T}}\mathbf{L}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}}. \tag{2.3}$$

The constraint $\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y} = 1$ removes an arbitrary scaling factor in the embedding. Since $\mathbf{L} = \mathbf{D} - \mathbf{W}$, the optimization problem in (2.3) has the following equivalent form

$$\mathbf{y}^* = \arg\max_{\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}=1} \mathbf{y}^{\mathrm{T}}\mathbf{W}\mathbf{y} = \arg\max \frac{\mathbf{y}^{\mathrm{T}}\mathbf{W}\mathbf{y}}{\mathbf{y}^{\mathrm{T}}\mathbf{D}\mathbf{y}}. \tag{2.4}$$

Assume that $\mathbf{y}$ is computed from a linear projection $\mathbf{y} = \mathbf{X}^{\mathrm{T}}\boldsymbol{v}$, where $\boldsymbol{v}$ is the unitary projection vector, (2.4) becomes

$$\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}=1} \boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}}\boldsymbol{v} = \arg\max \frac{\boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}}. \tag{2.5}$$

The optimal $\boldsymbol{v}$'s can be computed by solving the generalized eigenvalue decomposition problem

$$\mathbf{X}\mathbf{W}\mathbf{X}^{\mathrm{T}}\boldsymbol{v} = \lambda\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}. \tag{2.6}$$

LPP and NPE can be interpreted in this framework with different choices of $\mathbf{W}$ and $\mathbf{D}$ [9]. A brief explanation about the choices of $\mathbf{W}$ and $\mathbf{D}$ for LPP and NPE is provided in the following subsections.

### 2.1. Locality Preserving Projection (LPP)

LPP optimally preserves the neighbourhood structure of data set based on a heat kernel nearest neighbour graph [5]. Specifically, let $N_k(\mathbf{x}_i)$ denote the $k$ nearest neighbours of $\mathbf{x}_i$, $\mathbf{W}$ and $\mathbf{D}$ of LPP are denoted as $\mathbf{W}^{\mathrm{LPP}}$ and $\mathbf{D}^{\mathrm{LPP}}$, respectively, in such that,

$$W_{ij}^{\mathrm{LPP}} = \begin{cases} \exp\left(-\dfrac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right), & \text{if } \mathbf{x}_i \in N_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in N_k(\mathbf{x}_i), \\ 0, & \text{otherwise} \end{cases} \tag{2.7}$$

and $D_{ii}^{\mathrm{LPP}} = \sum_j W_{ji}^{\mathrm{LPP}}$, which measures the local density around $\mathbf{x}_i$. The reader is referred to [5] for details.

### 2.2. Neighbourhood Preserving Embedding (NPE)

NPE takes into account the restriction that neighbouring points in the high-dimensional space must remain within the same neighbourhood in the low-dimensional space. Let $\mathbf{M}$ be a $n \times n$ local reconstruction coefficient matrix. For $i$th row of $\mathbf{M}$, $M_{ij} = 0$ if $\mathbf{x}_j \notin N_k(\mathbf{x}_i)$ where $N_k(\mathbf{x}_i)$ represents the $k$ nearest neighbours of $\mathbf{x}_i$. Otherwise, $M_{ij}$ can be computed by minimizing the following objective function

$$\min \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in N_k(\mathbf{x}_i)} M_{ij}\mathbf{x}_j \right\|^2, \qquad \sum_{\mathbf{x}_j \in N_k(\mathbf{x}_i)} M_{ij} = 1. \tag{2.8}$$

$\mathbf{W}$ and $\mathbf{D}$ of NPE are denoted as $\mathbf{W}^{\mathrm{NPE}}$ and $\mathbf{D}^{\mathrm{NPE}}$, respectively, where $\mathbf{W}^{\mathrm{NPE}} = \mathbf{M} + \mathbf{M}^{\mathrm{T}} - \mathbf{M}^{\mathrm{T}}\mathbf{M}$ and $\mathbf{D}^{\mathrm{NPE}} = \mathbf{I}$. Refer to [7] for the detailed derivation.

## 3. Eigenvector Weighting Function

Since $\mathbf{y} = \mathbf{X}^{\mathrm{T}}\boldsymbol{v}$, (2.3) becomes

$$\boldsymbol{v}^* = \underset{\boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}=1}{\arg\min} \boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\boldsymbol{v} = \arg\min \frac{\boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{L}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\mathbf{X}\mathbf{D}\mathbf{X}^{\mathrm{T}}\boldsymbol{v}}. \tag{3.1}$$
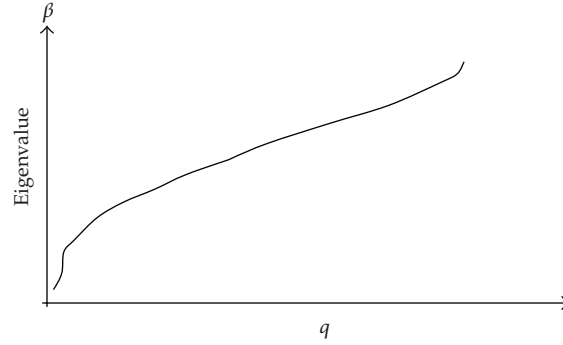
**Figure 1:** A typical eigenspectrum.

The optimal $\boldsymbol{v}$'s are the eigenvectors of the generalized eigenvalue decomposition problem associated with the smallest eigenvalues $\beta$'s

$$\mathbf{XLX}^{\mathrm{T}}\boldsymbol{v} = \beta\mathbf{XDX}^{\mathrm{T}}\boldsymbol{v} \quad \text{where } \beta = 1 - \lambda. \tag{3.2}$$

Cai et al. defined the locality preserving capacity of a projection $\boldsymbol{v}$ as [10]:

$$f(\boldsymbol{v}) = \frac{\boldsymbol{v}^{\mathrm{T}}\mathbf{XLX}^{\mathrm{T}}\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\mathbf{XDX}^{\mathrm{T}}\boldsymbol{v}}. \tag{3.3}$$

The smaller the value of $f(\boldsymbol{v})$ is, the better the locality preserving capacity of the projection $\boldsymbol{v}$. Furthermore, the locality preserving capacity has a direct relation to the discriminating power [10]. Based on the Rayleigh quotient form of (3.2), $f(\boldsymbol{v})$ in (3.3) is exactly the eigenvalue in (3.2) corresponding to eigenvector $\boldsymbol{v}$. Hence, the eigenvalues $\beta$'s reflect the data locality. The eigenspectrum plot of $\beta$ against the index $q$ is a monotonically increasing function as shown in Figure 1.

### 3.1. Eigenspace Decomposition

In graph-based subspace learning approach, local geometrical structure of data is defined by the assigned neighbourhood. Without any prior information about class label, the neighbourhood, $N_k(\mathbf{x}_i)$ is selected blindly in such a way that neighbourhood is simply determined by the $k$ nearest samples of $\mathbf{x}_i$ from any classes. If there are large within-class variations, $N_k(\mathbf{x}_i)$ may not be from the same class of $\mathbf{x}_i$; and, the algorithm will include them to characterize the data properties, in which lead to undesirable recognition performance.

To inspect the empirical eigenspectrum of graph-based subspace learning approach, we take 300 facial images of 30 subjects (10 images per subject) from Essex94 database [11] and 360 images of 30 subjects (12 images per subject) from FRGC face database [12] to render eigenspectra of NPE and LPP. The images in Essex94 database for a particular subject are similar in such a way that there are very minor variations in head turn, tilt and slant, as well as very minor facial expression changes as shown in Figure 2. Besides, there is no changing in terms of head scale and lighting. In other words, Essex94 database is simpler with minimum

**Figure 2:** Five face image samples from the Essex94 database.



**Figure 3:** Five face image samples from the FRGC database.

intraclass variation. On the other hand, FRGC database appears to be more difficult due to variations of scale, illumination and facial expressions as shown in Figure 3.

Figures 4 and 5 illustrates the eigenspectra of NPE and LPP. For better illustration, we zoom into the first 40 eigenvalues, as shown in part (b) of each figure. We observe that the first 20 NPE-eigenvalues in Essex94 are zero, but not for FRGC. Similar result is found in LPP. The reason is that the facial images of Essex94 of a particular subject are nearly identical, which imply low within-class variations in the images cause better neighbourhood selection for defining local geometrical properties, leading to high data locality. On the other hand, images of FRGC are of vary due to large intraclass variations, thus lower data locality is obtained due to inadequate neighbourhood selection. For practical face recognition without controlling the environmental factors, the intravariations of a subject are inevitably large due to different poses, illumination and facial expressions. Hence, the first portion of the eigenspectrum spanned by $q$ eigenvectors corresponding to the first $q$ smallest eigenvalues is marked as noise I subspace (denoted as **N-I**).

Eigenfeatures that are extracted by graph-based subspace learning approach are noise prompted due to external factors, such as sensors, unknown noise sources, and so forth, which will affect the recognition performance. From the empirical results shown in Figure 6, it is observed that after $q = 40$, recognition error rate increased for Essex94; and no further improvement in recognition performance on FRGC even $q > 80$ was considered. Note that the recognition error rate is average error rate (AER), which is the mean value of false accept rate (FAR) and false reject rate (FRR). The results demonstrated that the inclusion of eigenfeatures that correspond to large $\beta$ could be detrimental to recognition performance. Hence, we name this part as noise II subspace, denoted as **N-II**. The intermediate part between **N-I** and **N-II** is then identified as the intrinsic face dominated subspace, and denoted as **F**.

Since face images have similar structure, facial components are intrinsically resided in a very low-dimensional subspace. Hence, in this paper, we estimate the upper bound of the eigenvalues, $\beta$ that associated with face dominating eigenvectors is $\lambda_m$ where $m = \lfloor 0.25 * Q \rfloor$, where $Q$ is the total number of eigenvectors. Besides that, we assume the span of **N-I** is relatively small compared to **F**, in such a way that **N-I** is about 5% and **F** is about
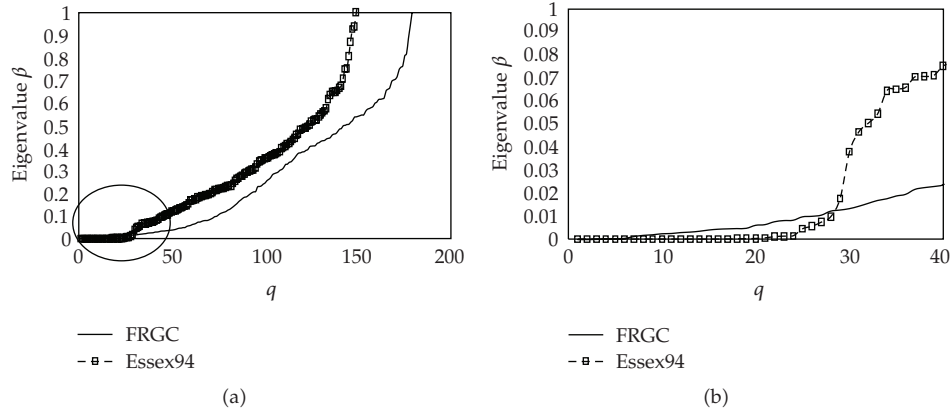
**Figure 4:** Typical real NPE-eigenspectra of (a) a complete set of eigenvectors and (b) the first $q$ eigenvectors.
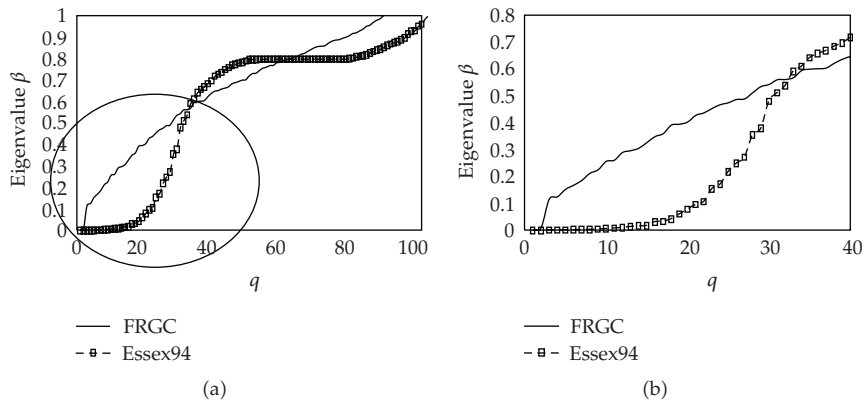


**Figure 5:** Typical real LPP- eigenspectra of (a) a complete set of eigenvectors and (b) the first $q$ eigenvectors.
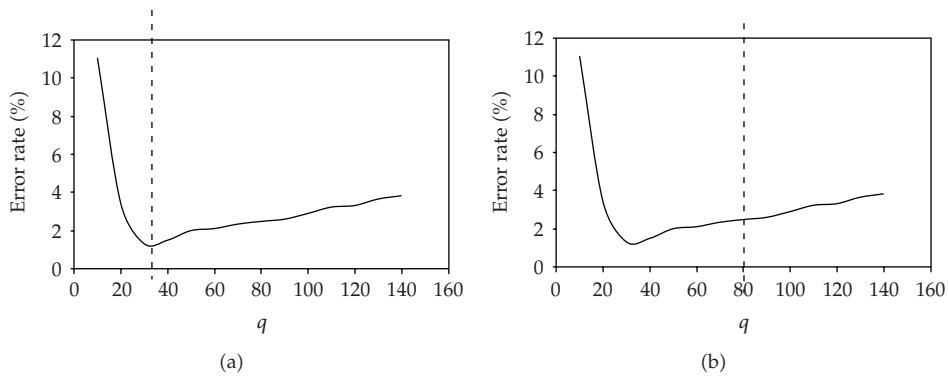


**Figure 6:** Recognition performances of NPE in term of average error rate on (a) Essex94 and (b) FRGC databases.
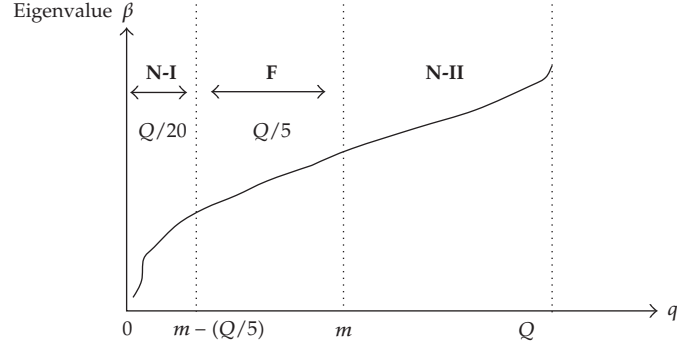
**Figure 7:** Decomposition of the eigenspace.

20% of the entire subspace. The subspace above $\lambda_m$ is considered as **N-II**. The eigenspace decomposition is illustrated in Figure 7.

### 3.2. Weighting Function Formulation

We devise a piecewise weighting function, coined as Eigenvector Weighting Function (EWF) to weight the eigenvectors differently in the decomposed subspaces. The principal of EWF is that larger weights will be imposed to the informative face dominating subspace, whereas smaller weighting factors are granted to the noise I and noise II subspaces to deemphasize the effect of the noisy eigenvectors in recognition performance. Since the eigenvectors in **N-II** contribute nothing to recognition performance, as validated in Figure 6, zero weight should be granted to the eigenvectors. Based on the principal, we propose a piecewise weighting function in such that weight values are increased from **N-I** to **F** and decreased from **F** to **N-II** until zero value to the remaining eigenvectors in **N-II**, refer to Figure 8. EWF is formulated as,

$$
w_q = \begin{cases} sq + c, & 1 \leq q \leq m - \left(\dfrac{Q}{5}\right), \\[2ex] w_{q_{m-(Q/5)}}, & m - \left(\dfrac{Q}{5}\right) < q \leq m, \\[2ex] -sq + c + \left(s\left(2m - \left(\dfrac{Q}{5}\right)\right)\right), & m < q \leq 2m - \left(\dfrac{Q}{5}\right), \\[2ex] 0, & q > 2m - \left(\dfrac{Q}{5}\right), \end{cases} \tag{3.4}
$$

where $s = (h-c)/(m-(Q/10)-1)$ is the slope of a line connecting from $(1, c)$ to $(m-(Q/10), h)$. In this paper, we set $h = 100$ and $c = 0.1$.
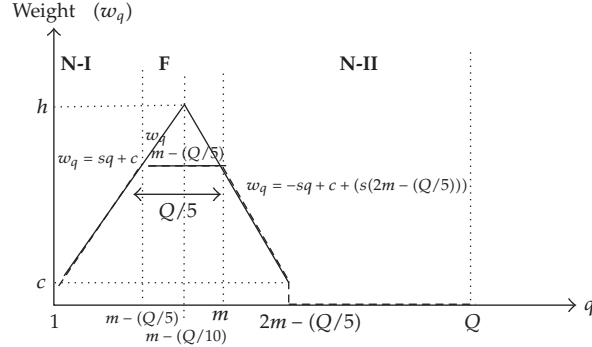
**Figure 8:** The weighting function of Eigenvector Weighting Function (EWF), represented in the dotted line.

## 3.3. *Dimensionality Reduction*

New image data $\mathbf{x}_i$ is transformed into lower-dimensional representative vector $\mathbf{y}_i$ via a linear projection as shown below

$$\mathbf{y}_i = \widetilde{\boldsymbol{\nu}}^{\mathrm{T}}\mathbf{x}_i, \tag{3.5}$$

where $\widetilde{\boldsymbol{\nu}}$ is the set of regularized projection directions, $\widetilde{\boldsymbol{\nu}} = [\mathrm{w}_i\boldsymbol{\nu}_i]_{i=1}^{Q} = [\mathrm{w}_1\boldsymbol{\nu}_1, \ldots, \mathrm{w}_Q\boldsymbol{\nu}_Q]$.

## 4. Numerical Justification of EWF

In order to validate the effectiveness of the proposed weighting selection, we compare the recognition performance of EWF with other arbitrary weighting functions: (1) *InverseEWF*, (2) *Uplinear*, and (3) *Downlinear*. In contrast to EWF, *InverseEWF* imposes very small weights to **F** but emphasizes the noise I and II eigenvectors by decreasing the weights from **N-I** to **F**, while increasing the weights from **F** to **N-II**. The *Uplinear* weighting function increases linearly while the *Downlinear* weighting function decreases linearly. Figure 9 illustrates the weighting scaling of EWF and the three arbitrary weighting functions.

Without loss of generality, we use NPE for the evaluation. The NPE with the above mentioned weighting functions are denoted as EWF_NPE, *InverseEWF*_NPE, *Uplinear*_NPE and *Downlinear*_NPE. In this experiment, a 30-class sample of FRGC database is adopted. From Figure 10, we observe that EWF_NPE outperforms the other weighting functions. By imposing larger weights to the eigenvectors in **F**, both EWF_NPE and *Uplinear*_NPE achieve lower error rates with small feature dimensions. Besides, the performance of *Uplinear*_NPE deteriorates in higher feature dimensions. The reason is that the emphasis of **N-II** eigenvectors leads to noise enhancement in this subspace.

Both *InverseEWF*_NPE and *Downlinear*_NPE emphasize **N-I** subspace and suppress the eigenvectors in **F**. These weighting functions have negative effects on the original NPE as illustrated in Figure 10. Specifically, *InverseEWF*_NPE ignores the significance of the face dominating eigenvectors by enforcing very small weighting factor (nearly zero weight) to the entire **F**. Hence, *InverseEWF*_NPE consistently shows the worst recognition performance for all feature dimensions. In Section 5, we investigate further the performance of the EWF for NPE and LPP using different face databases with larger sample size.
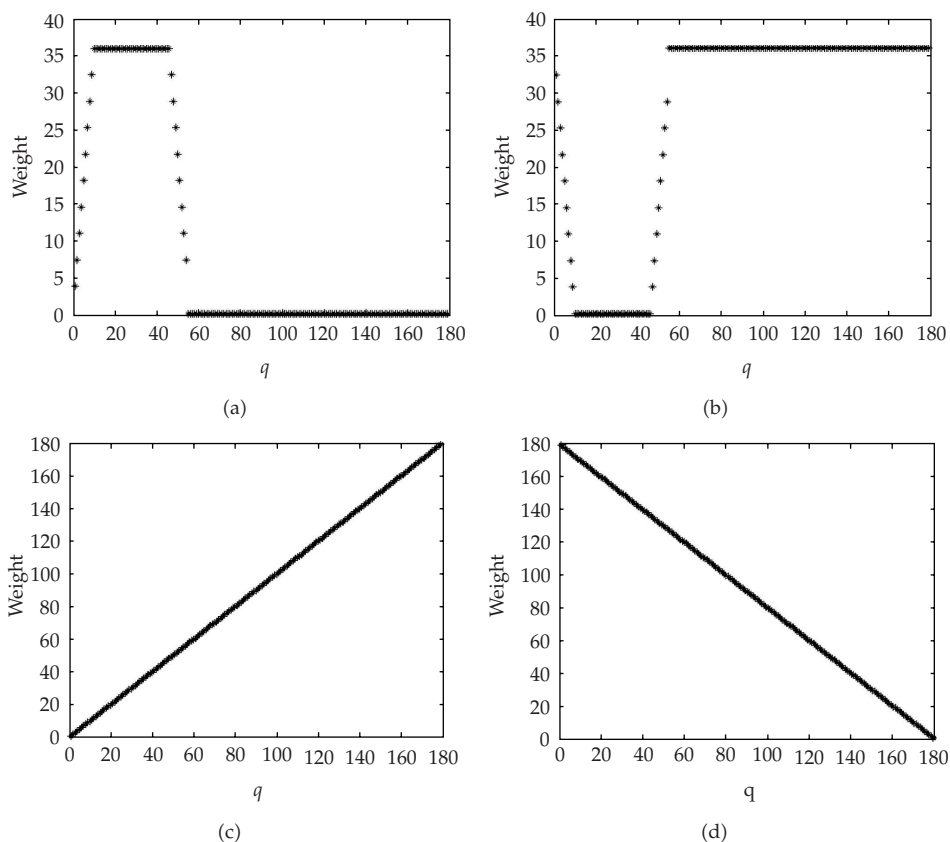
**Figure 9:** Different weighting functions: (a) the proposed EWF, (b) *InverseEWF*, (c) *Uplinear*, and (d) *Downlinear*.

## 5. Experimental Results and Discussions

In this section, EWF is applied to two graph-based subspace learning techniques: NPE and LPP, denoted as EWF_NPE and EWF_LPP, respectively. The effectiveness of EWF_NPE and EWF_LPP are assessed by two considerably difficult face databases: (1) Face Recognition Grand Challenge Database (FRGC) and (2) Face Recognition Technology (FERET) database. The FRGC data was collected at the University of Notre Dame [12]. It contains controlled images and uncontrolled images. The controlled images were taken under a studio setting. The images are full frontal facial images taken under two lighting conditions (two or three studio lights) and with two facial expressions (smiling and neutral). The uncontrolled images were taken under varying illumination conditions, for example, hallways, atria, or outdoors. Each set of uncontrolled images contains two expressions, smiling and neutral. In our experiments, we use a subset from both controlled and uncontrolled sets and randomly assign as training and testing sets. Our experimental database consists of 140 subjects with 12 images per subject. There is no overlapping between the images of this subset database and those of the 30-class sample database used in Section 4. The FERET images were collected for about three years, between December 1993 and August 1996, managed by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) [13]. In our experiments, a subset of this database is used, comprising 150
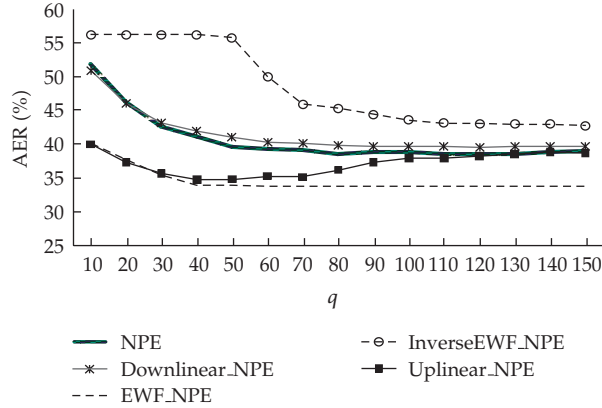
**Figure 10:** Recognition error rates of NPE, EWF_NPE, *Uplinear*_NPE, *Downlinear*_NPE and *Inverse-EWF*_NPE.



**Figure 11:** Five face image samples from the FERET database.

subjects with 10 images per subject. Five sample images from the FERET database are shown in Figure 11.

These images are preprocessed by using geometrical normalization in order to establish correspondence between face images. The procedure is based on automatic location of the eye positions, from which various parameters (i.e., rotation, scaling and translation) are used to extract the central part of the face from the original image. The database images are normalized into a canonical format. We apply a simple nearest neighbour classifier for sake of simplicity. The Euclidean metric is used as distance measure. Since the proposed approach is an unsupervised method, to have a fair performance comparison, it is tested and compared with the other unsupervised feature extractors, such as Principal Component Analysis (PCA) [14], NPE and LPP. The qualities of the feature extraction algorithms are evaluated in term of average error rate (AER).

For each subject, we randomly select $n_j$ samples and they are partitioned into training and testing sets with $n_j/2$ samples for each. Both training and testing sets have no overlap in the sample images between the training and testing sets. We conduct experiment with a 4-fold cross-validation strategy. In the first-fold test, the odd numbered images of each subject ($n_j/2$ samples per subject) are served as training images, while the even numbered images ($n_j/2$ samples per subject) are used as testing images. In the second-fold test, the even numbered images ($n_j/2$ samples per subject) are training set and the odd numbered images ($n_j/2$ samples per subject) are testing set. In the third-fold test, the first $n_j/2$ samples per subject are used for training and the rest are for testing. For forth-fold test, the training set is

**Table 1:** Details of FRGC and FERET databases.

| Database | Number of subjects, $c$ | Number of samples per subject, $n_j$ | Number of training samples, $n_j/2$ | Number of testing samples, $n_j/2$ |
|---|---|---|---|---|
| FRGC | 140 | 12 | 6 | 6 |
| FERET | 150 | 10 | 5 | 5 |
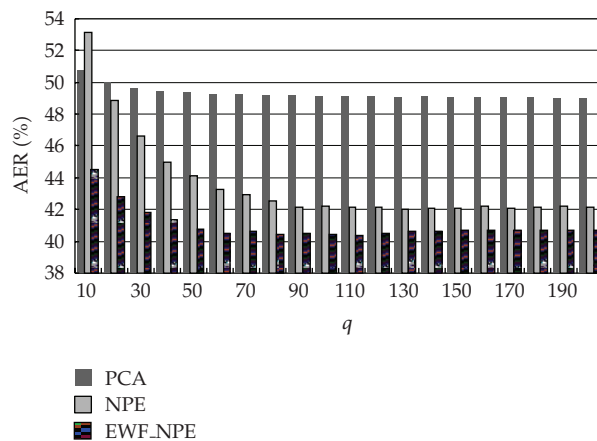


(a)
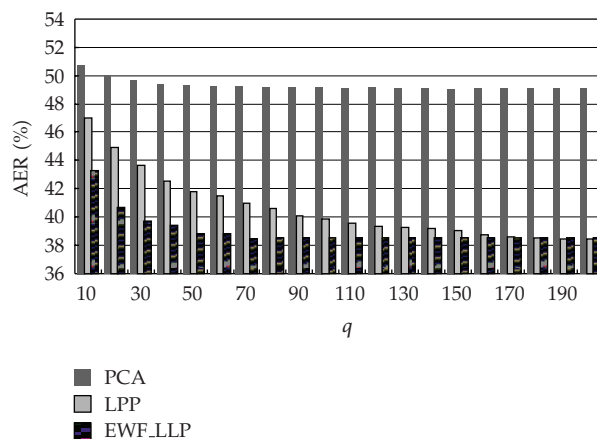


(b)

**Figure 12:** Error rates (%) of (a) PCA, NPE and EWF_NPE, (b) PCA, LPP and EWF_LPP on FRGC database.

formed by the last $n_j/2$ samples per subject and the rest are for testing. Table 1 summarizes the details of each database.

We set $N_k(\mathbf{x}_i) = (n_j/2) - 1$, that is, $N_k(\mathbf{x}_i) = 5$ and $N_k(\mathbf{x}_i) = 4$ on FRGC and FERET, respectively, for EWF_NPE, EWF_LPP, NPE and LPP. Besides, we evaluate the effectiveness of the techniques with different parameter settings. The ranges of the parameters are shown in Table 2. PCA ratio is the percentage of principal component kept in the PCA step and $\sigma$ indicates the spread of the heat kernel. The optimal parameter settings based on the empirical results are illustrated in Table 2. These parameter settings will be used in our subsequent experiments.
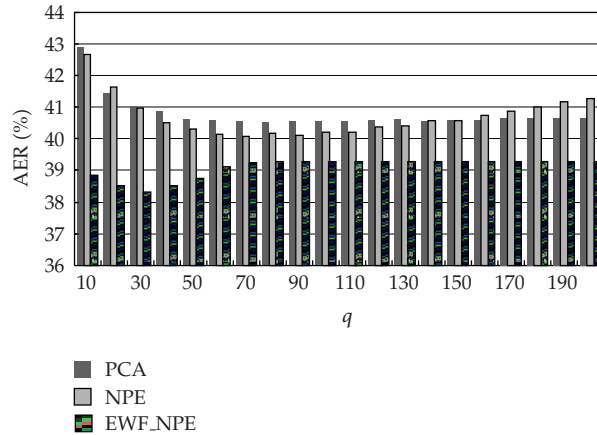
(a)
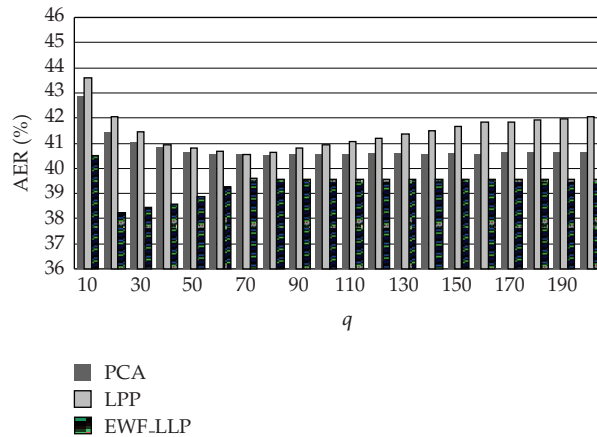


(b)

**Figure 13:** Error rates (%) of (a) PCA, NPE and EWF_NPE, (b) PCA, LPP and EWF_LPP on FERET database.

**Table 2:** Parameter ranges used in the experiments.

| Methods | Parameters | Parameter ranges | Optimal parameter settings |
|---------|-----------|------------------|---------------------------|
| NPE | PCA ratio | 98% and 100% | PCA ratio = 100% in FRGC; PCA ratio = 98% in FERET |
| EWF_NPE | PCA ratio | 98% and 100% | PCA ratio = 100% in FRGC; PCA ratio = 98% in FERET |
| LPP | PCA ratio | 98% and 100% | PCA ratio = 98% in FRGC and FERET |
| | $\sigma$ | 1, 10 and 100 | $\sigma$ = 1 in FRGC $\sigma$ = 100 in FERET |
| EWF_LPP | PCA ratio | 98% and 100% | PCA ratio = 98% in FRGC and FERET |
| | $\sigma$ | 1, 10 and 100 | $\sigma$ = 1 in FRGC $\sigma$ = 100 in FERET |

**Table 3:** Performance comparison in terms of average error rate.

| Approach | Feature dimension | Error rate of 1st fold test (%) | Error rate of 2nd fold test (%) | Error rate of 3rd fold test (%) | Error rate of 4th fold test (%) | Average error ± std deviation (%) |
|---|---|---|---|---|---|---|
| FRGC database | | | | | | |
| PCA | 200 | 52.624 | 51.569 | 44.681 | 47.155 | 49.007 ± 3.73 |
| NPE | 130 | 45.587 | 43.078 | 39.062 | 40.378 | 42.026 ± 2.90 |
| EWF_NPE | 110 | 44.409 | 40.424 | 36.969 | 39.686 | 40.372 ± 3.07 |
| LPP | 190 | 41.127 | 38.063 | 38.957 | 35.679 | 38.456 ± 2.25 |
| EWF_LPP | 70 | 41.300 | 38.919 | 36.684 | 36.621 | 38.381 ± 2.22 |
| FERET database | | | | | | |
| PCA | 80 | 46.823 | 38.971 | 31.514 | 44.811 | 40.530 ± 6.87 |
| NPE | 70 | 44.169 | 37.883 | 32.538 | 45.737 | 40.082 ± 6.06 |
| EWF_NPE | 30 | 43.644 | 37.998 | 33.029 | 38.561 | 38.308 ± 4.33 |
| LPP | 70 | 44.631 | 38.657 | 32.516 | 46.358 | 40.541 ± 6.28 |
| EWF_LPP | 20 | 43.351 | 36.924 | 32.948 | 39.719 | 38.235 ± 4.39 |

PCA is a global technique that analyzes image as a whole data matrix. Technically, PCA relies on sample data to compute total scatters. On the other hand, NPE and LPP signify the intrinsic geometric structure and extract the discriminating features for data learning. Hence, NPE and LPP outperform PCA on the FRGC database as demonstrated in Figure 12. However, the good recognition performance of both graph-based methods is not guaranteed when applied to the FERET database. From Figure 13, NPE and LPP show inferior performance compared to PCA when small feature dimension as well as large feature dimension is considered. The unreliable features at the lower order and higher order eigenvectors could be the factor for the performance degradation.

From Figures 12 and 13, we observe that EWF_NPE and EWF_LPP achieve lower error rate than their counterpart at smaller feature dimension on both databases. This implies that the strategy of penalizing the eigenvectors in **N-I** and emphasizing the face dominating eigenvectors in **F** is promising. Furthermore, the robustness of EWF can be further validated through the recognition results of FERET database. In FERET database, even though both NPE and LPP do not perform in the higher feature dimension, EWF_NPE and EWF_LPP consistently demonstrate better results due to small or zero weighting on eigenvectors in **N-II**.

Table 3 shows the average error rates, as well as the standard deviation of the error, on FRGC and FERET databases. The table summarizes the recognition performances along with the subspace dimension corresponding to the best recognition. In FRGC database, EWF shows its robustness in face recognition when implemented in NPE algorithm. Besides, we can see that the performance of EWF_LPP is comparable to that of LPP. However, the former is able to reach the optimal performance with smaller number of features. On the other hand, both EWF_NPE and EWF_LPP outperform their counterparts (NPE and LPP) on FERET database. Furthermore, they achieve such good performance with smaller number of features.

## 6. Conclusion

We have presented an eigenvector weighting function (EWF) and implemented it on two graph-based subspace learning techniques: Locality Preserving Projection and Neighbourhood Preserving Embedding. In EWF, the eigenspace of the learning approach is decomposed into three subspaces: (1) a subspace due to facial intraclass variations, (2) an intrinsic face subspace, and (3) a subspace that is attributed to sensor and external noises. Then, weights are imposed to each subspace differently. It grants higher weighting to the face variation dominating eigenvectors, while demphasizing the other two noisy subspaces with smaller weights. The robustness of EWF is assessed in two graph-based subspace learning techniques: Locality Preserving Projection (LPP) and Neighbourhood Preserving Embedding (NPE) on FRGC and FERET databases. The experimental results exhibit the robustness of the proposed EWF in face recognition.

## References

[1] K. Fukunnaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 2nd edition, 1991.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[4] Y. Fu, S. Yan, and T. S. Huang, "Classification and feature extraction by simplexization," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 91–100, 2008.

[5] X. He, S. Yan, Y. Hu, P. Niyogi, and H. J. Zhang, "Face recognition using laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[6] E. A. Eimad and E. K. Wong, "Locally linear discriminate embedding for face recognition," *Discrete Dynamics in Nature and Society*, vol. 2009, Article ID 916382, 8 pages, 2009.

[7] X. He, D. Cai, S. Yan, and H. J. Zhang, "Neighborhood preserving embedding," in *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV '05)*, pp. 1208–1213, October 2005.

[8] S. Yan, D. Xu, B. Zhang, H. J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[9] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–7, June 2007.

[10] D. Cai, X. He, J. Han, and H. J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608–3614, 2006.

[11] Essex 94 Face Database, http://cswww.essex.ac.uk/mv/allfaces/index.html.

[12] P. J. Phillips, P. J. Flynn, T. Scruggs et al., "Overview of the face recognition grand challenge," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 947–954, June 2005.

[13] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 137–143, 1997.

[14] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.