Graph-Preserving Sparse Non-Negative Matrix Factorization with Application to Facial Expression Recognition

Ruicong Zhi, Markus Flierl, *Member, IEEE* Qiuqi Ruan, *Senior Member, IEEE* and W. Bastiaan Kleijn, *Fellow, IEEE*

Abstract

In this paper, a novel graph-preserving sparse non-negative matrix factorization (GSNMF) algorithm is proposed for facial expression recognition. The GSNMF algorithm is derived from the original NMF algorithm by exploiting both sparse and graph-preserving properties. The latter may contain the class information of the samples. Therefore, GSNMF can be conducted as an unsupervised or a supervised dimension reduction method. A sparse representation of the facial images is obtained by minimizing the l^1-norm of the basis images. Furthermore, according to graph embedding theory, the neighborhood of the samples is preserved by retaining the graph structure in the mapped space. The GSNMF decomposition transforms the high-dimensional facial expression images into a locality-preserving subspace with sparse representation. To guarantee convergence, we use the projected gradient method to calculate the non-negative solution of GSNMF. Experiments are conducted on the JAFFE database and the Cohn-Kanade database with not-occluded and partially occluded facial images. The results show that the GSNMF algorithm provides better facial representations and achieves higher recognition rates than NMF. Moreover, GSNMF is also more robust to partial occlusions than other tested methods.

Index Terms

Facial expression recognition, locality preservation, sparseness, non-negative matrix factorization.

R. Zhi and Q. Ruan are with the Institute of Information Science, Beijing Jiaotong University, Beijing 100044 , P. R. China. e-mail: 05120370@bjtu.edu.cn. M. Flierl and W. B. Kleijn are with the ACCESS Linnaeus Center, School of Electrical Engineering, KTH – Royal Institute of Technology, Stockholm.

This work has been completed while R. Zhi visited KTH. It has been supported by the Technological Innovation Fund for Excellent Doctoral Candidate of Beijing Jiaotong University (48036, 141037522), the National Natural Science Foundation of China (No. 60973060), the Doctorial Foundation of Ministry of Education of China (No. 200800040008), and the Project of Educational Committee of Beijing (No. YB20081000401).

I. INTRODUCTION

In recent years, the demand for human-computer interaction has increased significantly. Facial expression recognition is one of the most important subjects in the fields of human-computer interaction and information processing [1]. As a reflection of emotions, facial expressions have been extensively studied in psychology [2][3][4]. The study of Mehrabian [5] indicated that in face-to-face communication of human, 7% of the communication information is transferred by linguistic language, 38% by paralanguage, and 55% by facial expressions. This indicates that facial expression plays an important role in human communication.

It is commonly accepted that the intrinsic dimensionality of the space of possible face images is much lower than that of the original image space. Thus, it is necessary to look for efficient dimensionality reduction methods for facial feature extraction. Also, good features for representing facial expression can alleviate the complexity of the classification algorithm design. Many techniques have been proposed to analyze facial expressions [6][7], and the most popular class of methods are subspace-based algorithms. Basically, the subspace-based algorithms discover the latent facial features by decomposing (projecting) the image onto a linear (nonlinear) low dimensional image subspace. There are many commonly used subspace-based methods, e.g. Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Locality Preserving Projections (LPP). PCA [8] represents the faces by projecting the facial images to the directions of maximal covariance in the facial image data. The basis images corresponding to PCA are ordered according to the decreasing amount of variance they represent, i.e. the respective eigenvalues. LDA [9] projects the images onto a subspace in such a way that the ratio of the between-class scatter and the within-class scatter is maximized. LDA is a class-specific projection method, generally outperforming PCA. LPP [10] aims to find an embedding that preserves local information, and obtains a face subspace that best detects the essential face manifold structure. These approaches have been shown to be efficient in recognizing facial expressions. Shan et.al [11] revealed the comprehensive comparison of some commonly used subspace methods for facial expression recognition, and concluded that supervised LPP outperforms the other supervised methods. In facial analysis problems, a researcher is often confronted with the singular problem that arises from the fact that image dimensions are extremely larger than the number of samples. A solution to this problem is to utilize PCA as a preprocessing step to reduce the dimensionality of the image space. Therefore, Eigenfaces (PCA), Fisherfaces (PCA plus LDA) [12] and Laplacianfaces (PCA plus LPP) [13] were developed for facial image processing.

The aforementioned algorithms result in holistic representations for facial images, which store the

facial image as a perceptual whole, without explicitly specifying its parts (e.g. eyes, nose, mouth, chin). They can also be seen as dense image representations [14]. On the contrary, other researches argue for a sparse image representation that leads to ”efficient coding” in the visual cortex [15]. Ellison and Massaro [16] revealed that facial expressions are better represented by facial parts, suggesting a non-holistic representation.

Non-negative matrix factorization (NMF) algorithm is a recent method for finding a non-negative decomposition of the original data matrix. It has been used for various applications [17][19]. NMF is based on the idea that negative numbers are physically meaningless in many data-processing tasks. NMF represents a facial image as a linear combination of basis images. The difference between NMF and PCA is that NMF guarantees non-negativity of the elements in both basis vectors and the representation weights used in the linear combination. Lee and Seung [18] showed that NMF can learn a parts-based representation. The basis images consist of basis vectors representing eyes, nose, mouse, etc. Although the decomposition tends to produce parts-based representations of basis images by composing the parts in an additive fashion, this is not always the case. Several work revealed that NMF decomposition often produces a holistic image representation [19][20][21]. Several extended NMF algorithms have been proposed to obtain a local representation of the facial images, e.g. LNMF (Local Non-negative Matrix Factorization) [21] and DNMF (Discriminant Non-negative Matrix Factorization) [22]. A comparison among these three algorithms for facial expression recognition indicates that DNMF outperforms LNMF and NMF [23].

In this paper, we develop a novel graph-preserving sparse non-negative matrix factorization (GSNMF) algorithm for facial expression recognition. A preliminary version of the algorithm was presented in [24]. This method is derived from the NMF algorithm and exploits both sparse and discriminant properties. First, we extend NMF to include a sparseness constraint which is measured by l^1-norm of the basis images obtained in the non-negative decomposition. Then, according to graph embedding theory, the neighborhood of the samples is preserved by minimizing the graph-preserving criterion in the mapped space [13]. Thus, we exploit locality preserving information in addition to the sparseness constraint. Furthermore, the similarity matrix in the graph-preserving constraint can involve information of various images separated to different facial expression classes to improve the classification performance. Therefore, the algorithm can be used in a supervised manner by considering the neighborhood information of the image data. Furthermore, we show that the DNMF algorithm can be incorporated into the GSNMF framework.

A widely used solution for NMF-based methods is the multiplicative update method, which was

proposed by Lee and Seung [25]. The NMF optimization problem is nonconvex. Algorithms usually deal with several local minima and produce sequences of iterations. A common misunderstanding is that the limit points of such sequences are local minima [25][26][27]. In optimization theory, a local minimum must be a stationary point. The commonly used multiplicative update method can not guarantee stationary and convergence [28]. In this work, we use the projected gradient method to ensure the stationarity of limit points. This method has been successfully used for the NMF algorithm [27]. We introduce a projected gradient framework for constrained NMF algorithms, and utilize the proposed framework to solve the GSNMF algorithm.

This paper is organized as follows: The motivation is outlined in Section II. The projected gradient framework for constrained NMF is described in Section III. The graph-preserving sparse non-negative matrix factorization (GSNMF) algorithm is proposed in Section IV; Experimental results that verify our approach for facial expression recognition are depicted in Section V. Finally, conclusions are drawn in Section VI.

II. Locality Preservation for NMF-based Algorithm

The key ingredient of NMF is the non-negative constraint imposed on the two decomposed matrices. The non-negative constraints are compatible with the intuitive notion of combining parts to form a whole. As a parts-based representation can naturally deal with partial occlusion and some illumination problems, it is considered to perform superior for facial image processing. Suppose there are n points $X = [x_1, x_2, \ldots, x_n]$ in a high-dimensional image space R^m, each image x_i denoted by a m-dimensional vector $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,m}]^T$. Thus, the image database is represented by a $m \times n$ matrix X. X can be decomposed into the product of two non-negative matrices, such that

$$X \approx WH, \tag{1}$$

where W is a $m \times p$ matrix, whose columns comprise basis functions and H is a $p \times n$ coefficient matrix. Usually p is chosen so that $(m + n)p < nm$ to achieve compression or dimensionality reduction. Each element x_{ij} of the matrix X can be written as $x_{ij} = \sum_k w_{ik} h_{kj}$. There are two common methods to measure the quality of the approximation. One is the square of the Euclidean distance

$$D(X||WH) = ||X - WH||_F^2 = \sum_{ij} \left(x_{ij} - \sum_k w_{ik} h_{kj} \right)^2. \tag{2}$$

The other is the Kullback-Leibler divergence between X and WH

$$D(X||WH) = \sum_{ij} \left(x_{ij} \ln \frac{x_{ij}}{\sum_k w_{ik} h_{kj}} + \sum_k w_{ik} h_{kj} - x_{ij} \right). \quad (3)$$

NMF aims to find the non-negative decomposition of X according to the following optimization problem:

$$\min D(X||WH)$$
$$\text{s. t. } w_{i,k} \geq 0, h_{k,j} \geq 0, \forall k, i, j \quad (4)$$

As NMF does not involve any class information, the method is not capable of taking advantage of discriminant aspects and subsequently generally performs poorly in terms of classification performance. To illustrate this point, let us consider 100 two-dimensional synthetic data points belonging to two classes, C_1 and C_2. The two classes are drawn from a two-dimensional Gaussian random variable. The two-class dataset is shown in Fig. 1(a). The data along with its one-dimensional NMF transformation are shown in Fig. 1(b) and the probability density functions (PDFs) of the projected data are shown in Fig. 1(c). The locality preserving power can be measured by $J = \sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||_2^2 S_{ij}$ [10], where $\tilde{x}_i$ and $\tilde{x}_j$ denote the projected points and S_{ij} are the elements of the similarity matrix. For NMF projected data, the value of locality preserving measure is 1246.5. Fig. 1(b) shows that the projected points overlap significantly. As the projected data also follow Gaussian probability distributions, the error rate of two-class classification can be calculated by

$$p(\text{error}) = \int_{-\infty}^{\Theta} p(\tilde{x}, C_1) d\tilde{x} + \int_{\Theta}^{+\infty} p(\tilde{x}, C_2) d\tilde{x}, \quad (5)$$

where $p(\tilde{x}, C_1)$, $p(\tilde{x}, C_2)$ are the joint PDFs of the projected data, and Θ is the decision threshold. This error is depicted as the overlapping area in Fig. 1(c).

As class information is important for classification problem, the locality structure of the data should be preserved in order to improve discriminant properties. By considering locality preserving properties, we introduce a graph-preserving constraint to the NMF algorithm and obtain an improved separation of the two classes. The projected data are shown in Fig. 2(a), and the corresponding Gaussian PDFs are shown in Fig. 2(b). It can be seen that the projected data are better separated when compared to NMF only. Consequently, the error rate (almost no overlap in Fig. 2(b)) is much smaller than that of NMF. The locality preserving measure of the projected data is 322.8 which is much smaller than that of NMF. This means that the classification results improve if the locality structure of the data is retained.

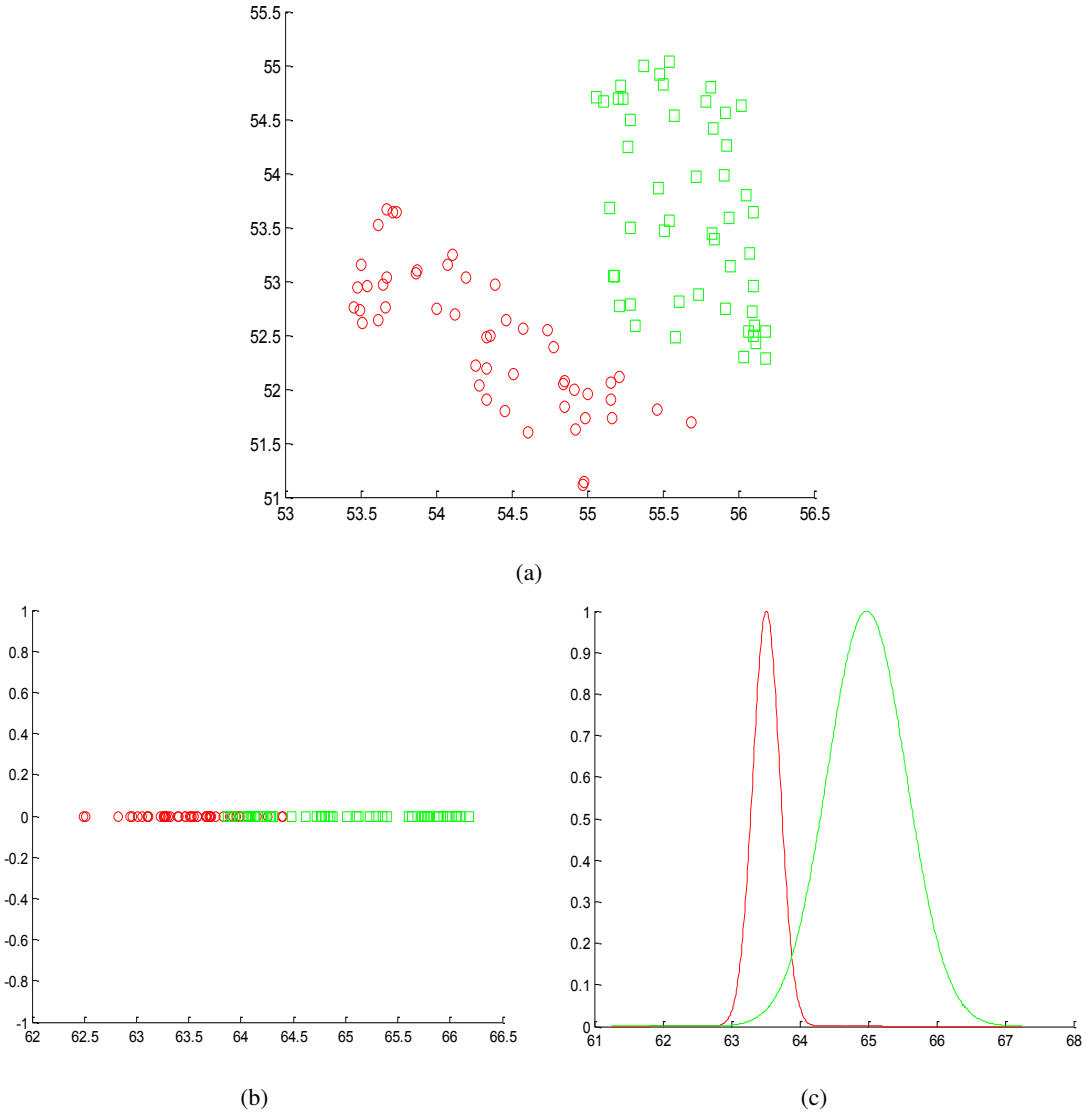
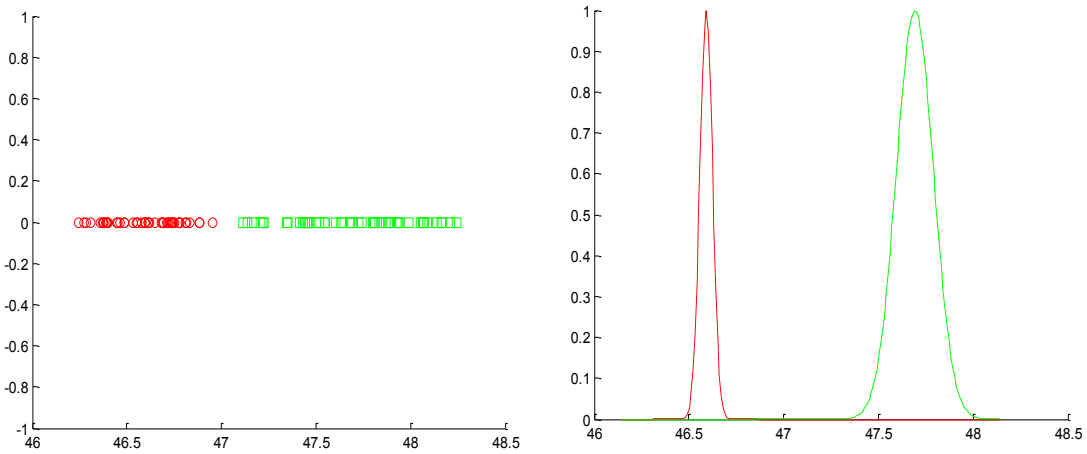


Fig. 1. NMF for a two-class problem: (a)original data set (b) reduced NMF decomposition (c) gaussian PDF of the reduced data.

III. Projected Gradient Framework

A popular approach to minimize the cost functions of various NMF algorithms is the multiplicative update method. In this method, the optimization of the cost function is performed using an auxiliary function. With the help of the auxiliary function, the optimal solution of W and H can be obtained by specified update rules, and the cost value is non-increasing after every update. However, it does not ensure the convergence of the algorithm to a limit point that is also a stationary point of the optimization problem



(a) (b)

Fig. 2. GSNMF for a two-class problem: (a) reduced GSNMF decomposition (b) gaussian PDF of the reduced data.

[26][28]. In order to ensure stationarity, we use projected gradients to solve the optimization problem with additional constraints. In this section, we give a projected gradient framework for constrained non-negative matrix factorization. The projected gradient method guarantees the stationarity of the limit points.

A. Constrained NMF Problem

The cost function of constrained non-negative matrix factorization can be stated as a Lagrangian formulation

$$D_{\text{cons}}(X||WH) = D(X||WH) + \zeta g(W,H), \tag{6}$$

where $g(W,H)$ is the constraint function with respect to W and H and is twice differentiable. ζ is a positive constant. The goal is to find W and H by solving the following problem:

$$\begin{aligned} &\min_{W,H} D_{\text{cons}}(X||WH) \\ &\text{s. t. } w_{i,k} \geq 0, h_{k,j} \geq 0, \forall k,i,j \end{aligned} \tag{7}$$

The optimization problem of (7) can be carried out by the block coordinate descent method for bound-constrained optimization [29]. We can fix one matrix and solve the optimization problem with respect to the other matrix. In order to find the solution, two functions are defined as $f_W(H) = D_{\text{cons}}(X||WH)$ and $f_H(W) = D_{\text{cons}}(X||WH)$ by keeping W and H fixed, respectively.

In order to find the optimal solution, the problem (7) is divided into two sub-problems: first, we fix H, and update W to achieve the conditional optimal value of the sub-minimization problem; second, we fix W, and update H to achieve the conditional optimal value of the sub-minimization problem.

B. Solving the Sub-Problems

We utilize the projected gradient method to solve the conditional optimization problems. When fixing H, the optimization with respect to W is an iterative procedure that is repeated until a stationary point is obtained. For the conditional problem $W^{(t+1)} = \arg\min_{W \geq 0} f_{H^t}(W)$, the update rule [27] is defined as

$$W^{(t+1)} = \left[W^{(t)} - \alpha_t \nabla f_H(W^{(t)}) \right]^+, \tag{8}$$

Equation (8) is a function of α_t, so we consider the following form

$$W^{(t+1)} = W^{(t)}(\alpha_t) = [W^{(t)} - \alpha_t \nabla f_H(W^{(t)})]^+, \tag{9}$$

where t is the number of iterations, $\alpha_t = \beta^{\varphi_t}$, so $W^{(t+1)} = W^{(t)}(\beta_t)$. φ_t is the first non-negative integer such that

$$\begin{aligned} f_H(W^{(t+1)}) - f_H(W^{(t)}) &\leq \sigma \left\langle \nabla f_H(W^{(t)}), W^{(t+1)} - W^{(t)} \right\rangle \\ \Rightarrow f_H(W^{(t)}(\beta^{\varphi_t})) - f_H(W^{(t)}) &\leq \sigma \left\langle \nabla f_H(W^{(t)}), W^{(t)}(\beta^{\varphi_t}) - W^{(t)} \right\rangle, \end{aligned} \tag{10}$$

The rule $[\cdot]^+ = \max[\cdot, 0]$ guarantees that the update does not contain any negative entries. The operator $\langle \cdot, \cdot \rangle$ is the Frobenius inner product between matrices. In our experiments, β and σ are chosen to be 0.1 and 0.01, respectively. To reduce the computational cost, inequality (10) can be reformulated as

$$(1-\sigma) \left\langle \nabla f_H(W^{(t)}), W^{(t+1)} - W^{(t)} \right\rangle + \frac{1}{2} \left\langle W^{(t+1)} - W^{(t)}, \left(W^{(t+1)} - W^{(t)} \right) \nabla^2 f_H(W^{(t+1)}) \right\rangle \leq 0. \tag{11}$$

The theoretical proof of the inequality (11) is in the Appendix.

C. Check of Stationarity

The iteration will stop when the solution is close to a stationary point. For bound-constrained optimization problems, a common condition to check if a point is close to a stationary point is [27]

$$||\nabla^P f_H(W^{(t)})||_F \leq \varepsilon_W ||\nabla f_H(W^{(1)})||_F, \tag{12}$$

where $||\cdot||_F$ is the Frobenius norm and $\nabla^P f_H(W)$ is the projected gradient defined as

$$\nabla^P f_H(W) = \begin{cases} [\nabla f_H(W)]_{i,k} & , \quad \text{if} \quad w_{i,k} > 0 \\ \min(0, [\nabla f_H(W)]_{i,k}) & , \quad \text{if} \quad w_{i,k} = 0, \end{cases} \tag{13}$$

where ε_W is the stopping tolerance which is predefined.

The procedure of solving the sub-problem of optimizing the cost function with respect to H for fixed W is similar with the discussion above. The two sub-problems are solved iteratively until the convergence rule is satisfied

$$||\nabla^P f_H(W^{(t)})||_F + ||\nabla^P f_W(H^{(t)})||_F \leq \varepsilon \left(||\nabla f_H(W^{(1)})||_F + ||\nabla f_W(H^{(1)})||_F \right). \tag{14}$$

IV. Graph-Preserving Sparse Non-negative Matrix Factorization (GSNMF) algorithm

In this section, we first discuss the use of the l^1-norm constraint for NMF in order to obtain a sparse representation of the facial images. Then locality-constraint optimization based on graph embedding is proposed. Finally, we obtain the graph-preserving sparse non-negative matrix factorization by considering the class information of the samples. GSNMF offers a sparse solution with more discriminant power than the NMF mehtod.

A. Sparse Solution via l^1-norm Constraint

For NMF, the columns of matrix W denote basis images and the elements of coefficient matrix H are non-negative. This means that we can only add the basis images rather than subtract to approximate the original image. For facial images, the basis images are supposed to be facial parts. However, NMF can not always give an intuitive decomposition into parts. This motivates the search for a parts-based representation of images. One of the solutions is to introduce sparseness constraint to NMF [20][30]. The concept of sparseness refers to a representation where most elements take values close to zero while only a few have significantly non-zero values.

The sparseness degree of a representation can be measured by the number of nonzero elements in the decomposition. It can be expressed by the l^0-norm, which counts the number of nonzero entries in a matrix. However, the problem of finding the sparsest solution of an underdetermined problem is hard to solve and even difficult to approximate [31]. The emerging theory of sparse representation and

compressed sensing [32][33] reveals that if the solution of the l^0-norm is sparse enough, the solution of the l^0-norm minimization problem is equal to the solution of the l^1-norm minimization problem. Therefore, the l^0-norm can be approximated by the l^1-norm. The l^1-norm is defined as $||W||_1 = \sum_{k,j} |w_{k,j}|$. By combining the goal of minimizing reconstruction error and sparseness, a new cost function is obtained.

$$\begin{aligned} D_{SNMF}(X||WH) &= D(X||WH) + \lambda \sum_{k,j} w_{k,j} \\ \text{s. t. } w_{i,k} \geq 0, h_{k,j} \geq 0, \textstyle\sum_i w_{i,k} &= 1, \forall k, i, j \end{aligned} \tag{15}$$

We use the square of the Euclidean distance to measure the error between X and WH. In this case, the cost function should become

$$D_{SNMF}(X||WH) = ||X - WH||_F^2 + \lambda \sum_{k,j} w_{k,j}, \tag{16}$$

where $||\cdot||_F$ is the Frobenius norm.

B. Derivation of Graph-preserving Sparse Non-negative Matrix Factorization (GSNMF)

In this subsection, we consider a supervised NMF-based dimensionality reduction method which is derived from the graph embedding analysis [13]. Let $G = \{X, S\}$ be an undirected weighted graph. X is the vector set, $X = [x_1, x_2, \cdots, x_n] \in \mathcal{R}^{m \times n}$ (for image processing problem, each entry of the matrix X denotes an image). $S \in \mathcal{R}^{n \times n}$ is the similarity matrix. Each element of the real symmetric matrix S measures the similarity between a pair of vertices, which is assumed to be non-negative in this work. There are various methods to form the similarity matrix, such as Gaussian kernel with Euclidean distance [34], local neighborhood relationship [35], and prior class information in supervised learning algorithms [36]. The Laplacian matrix L and the diagonal matrix E of the graph G are defined as

$$L = E - S, \quad E_{ii} = \sum_j S_{ij}. \tag{17}$$

We aim to find a map for the low-dimensional representation of the graph G. The similarity relationship between the vertex pairs are maintained in the corresponding low-dimensional subspace. Most NMF-based algorithms impose constraints on the coefficient matrix H, as H is the low-dimensional representation of original sample X. However, for pattern recognition tasks, the main purpose is to get good classification performance. For each original data point x_i, we project it by $\tilde{x}_i = W^T x_i$. The projected data matrix

$\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n] \in \mathcal{R}^{p \times n}$ is used for classification. Hence, we incorporate effective constraints on W. The graph-preserving criterion is given by

$$\sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||^2 S_{ij}. \tag{18}$$

The graph-preserving criterion aims to preserve the neighborhood of the data samples. If the similarity between points x_i and x_j is large, the distance between the corresponding projected points $\tilde{x}_i$ and $\tilde{x}_j$ should be small. If neighboring points are mapped far apart, then the criterion will incur a heavy penalty. The graph-preserving character is illustrated in Fig. 3. The graph-preserving criterion can be used unsupervised or supervised. If it is unsupervised, the similarity matrix S can be constructed according to the distances between pairs of samples, and the map will preserve the neighborhood of close samples, as shown in Fig. 3(a). If the criterion is supervised, the similarity matrix can be constructed utilizing the prior class information of the samples, and the map will preserve the class structure of the samples, as shown in Fig.3(b).

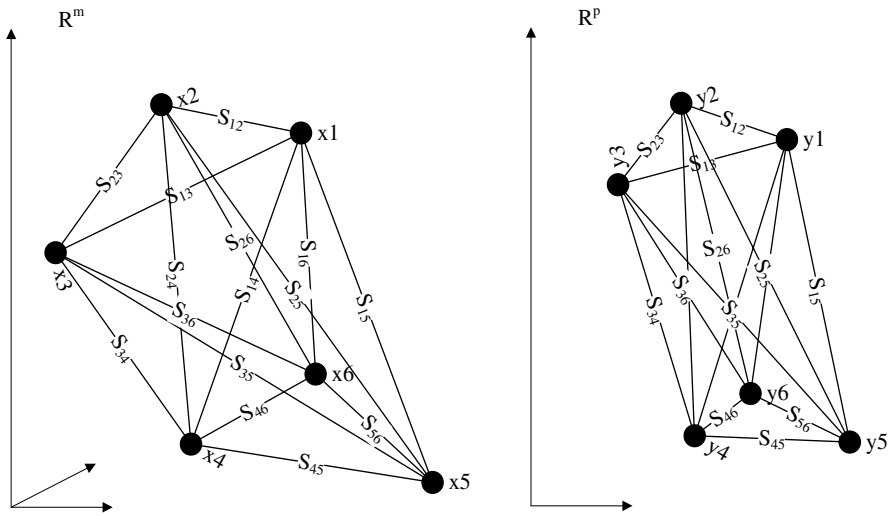
Now, we consider the non-negative decomposition problem which maps a graph into p-dimensional Euclidean space. The projective matrix is $W = [w_1, w_2, \cdots, w_p]$, where the columns of W provide the projective coordinates of the vertices. We need to find a tradeoff among reconstructive error, graph-preserving criterion, and sparseness. Thus, the cost function of graph-preserving sparse non-negative matrix factorization (GSNMF) is defined as

$$D_{GSNMF}(X||WH) = ||X - WH||_F^2 + \lambda \sum_{k,j} w_{k,j} + \eta \left(\sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||^2 S_{ij} \right), \tag{19}$$

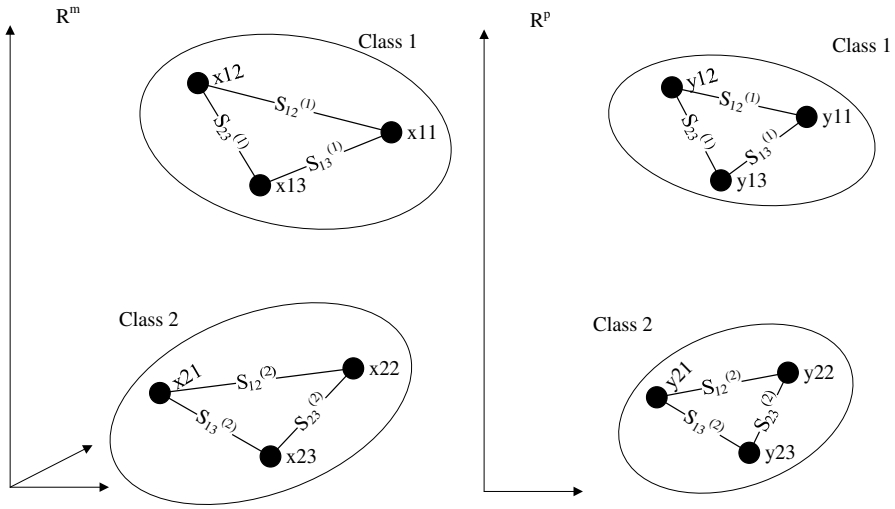
where $||\cdot||_F$ is the Frobenius norm, λ is a positive constant which controls the sparseness, and η is a positive constant which controls the locality of the decomposition. $\tilde{x}_i$ and $\tilde{x}_j$ are the data vectors after projecting to the image basis matrix W. The desired decomposition is obtained by solving the following optimization problem:

$$\begin{aligned} &\min_{W,H} D_{GSNMF}(X||WH) \\ &\text{s. t. } w_{i,k} \geq 0, h_{k,j} \geq 0, \textstyle\sum_i w_{i,k} = 1, \forall k, i, j \end{aligned} \tag{20}$$

The graph-preserving criterion can be rewritten as



(a) unsupervised case



(b) supervised case

Fig. 3. Graph of original data space and projected data space, y denotes projected points, which is $y = \tilde{x}_i = W^T x_i$.

$$
\begin{aligned}
\frac{1}{2}\sum_{i,j}||\tilde{x}_i - \tilde{x}_j||^2 S_{ij} &= \frac{1}{2}\sum_{i,j}\text{tr}\left[(\tilde{x}_i - \tilde{x}_j)(\tilde{x}_i - \tilde{x}_j)^T\right] S_{ij} \\
&= \frac{1}{2}\text{tr}\left[\sum_{i,j}(\tilde{x}_i - \tilde{x}_j)(\tilde{x}_i - \tilde{x}_j)^T S_{ij}\right] \\
&= \text{tr}\left[\sum_{i,j}\left(\tilde{x}_i S_{ij}(\tilde{x}_i)^T - \tilde{x}_i S_{ij}(\tilde{x}_j)\right)^T\right] \\
&= \text{tr}\left[\sum_{i}\tilde{x}_i E_{ii}(\tilde{x}_i)^T - \sum_{i,j}\tilde{x}_i S_{ij}(\tilde{x}_j)^T\right]
\end{aligned}
$$

$$
\begin{aligned}
&= \text{tr}\left[\sum_{i,j} \tilde{x}_i L_{ij} (\tilde{x}_j)^T\right] \\
&= \text{tr}(\tilde{X} L \tilde{X}^T), \qquad (21)
\end{aligned}
$$

where $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n]$ is a $p \times n$ matrix which is the data matrix after the projection on the image basis matrix W and the vector $\tilde{x}_i$ denotes the ith column of the matrix $\tilde{X}$. E is a diagonal matrix with entries that are column or row sums of S (since S is symmetric), i.e. $E_{ii} = \sum_j S_{ij}$. $L = E - S$ is the Laplacian matrix. In this work, we use a Gaussian kernel with Euclidean distance to construct the similarity matrix S. That is, if x_i and x_j belong to the same class, then $S_{ij} = \exp\left(||x_i - x_j||^2/t\right)$ (t is an empirical parameter); otherwise, $S_{ij} = 0$.

C. Projected Gradient Method for GSNMF

In order to obtain the solution of GSNMF algorithm, according to the analysis in Section III, we need to calculate the first and second order gradients of the two functions $f_W^{GSNMF}(H)$ and $f_H^{GSNMF}(W)$. The GSNMF cost function can be rewritten as $D_{GSNMF} = J_1 + \lambda J_2 + \eta J_3$, where $J_1 = ||X - WH||^2$, $J_2 = \sum_{k,j} w_{k,j}$ and $J_3 = \sum_{i,j} ||\tilde{x}_i - \tilde{x}_j||^2 S_{ij}$. Thus, the first- and second-order gradients of $f_W^{GSNMF}(H)$ and $f_H^{GSNMF}(W)$ are composed by the gradients of the three functions. J_1 is the function of both W and H, J_2 and J_3 are functions of W. Then $f_W^{GSNMF}(H)$ and $f_H^{GSNMF}(W)$ can be simplified as

$$
\begin{aligned}
f_W^{GSNMF}(H) &= J_1^H + C_1 \quad (C_1 \text{ is constant}) \\
f_H^{GSNMF}(W) &= J_1^W + \lambda J_2^W + \eta J_3^W. \qquad (22)
\end{aligned}
$$

Thus, the gradients of these two functions are

$$
\begin{aligned}
\nabla f_W^{GSNMF}(H) &= \nabla J_1^H \\
\nabla^2 f_W^{GSNMF}(H) &= \nabla^2 J_1^H \\
\nabla f_H^{GSNMF}(W) &= \nabla J_1^W + \lambda \nabla J_2^W + \eta \nabla J_3^W \qquad (23) \\
\nabla^2 f_H^{GSNMF}(W) &= \nabla^2 J_1^W + \lambda \nabla^2 J_2^W + \eta \nabla^2 J_3^W.
\end{aligned}
$$

We can write J_1^H as

$$\begin{aligned} J_1^H &= ||X - WH||_F^2 \\ &= \text{tr}\left((X - WH)^T(X - WH)\right) \\ &= \text{tr}(H^T W^T W H - H^T W^T X - X^T W H + X^T X). \end{aligned} \tag{24}$$

Both X and W are constant matrices. According to the matrix calculus, we get the first gradient of J_1^H as

$$\begin{aligned} \nabla J_1^H &= 2W^T W H - 2W^T X^T \\ &= 2W^T(WH - X). \end{aligned} \tag{25}$$

To calculate the second derivative, we use vector notation. If we concatenate the columns of H to a vector vec(H), then (24) can be written as

$$J_1^H = \text{vec}(H)^T \begin{bmatrix} W^T W & & \\ & \ddots & \\ & & W^T W \end{bmatrix} \text{vec}(H) + H\text{'s linear terms} + C_2, \tag{26}$$

where C_2 is constant. The second partial derivatives of J_1^H with respect to vec(H) form the Hessian matrix. From (26), we can see that the Hessian matrix is block diagonal, and each block $W^T W$ is a positive semi-definite matrix.

The function $f_H^{GSNMF}(W)$ consists of three parts. First, we use a similar procedure as explained above to obtain the first and second order gradients of J_1^W. The first order gradient of J_1^W is obtained by the partial derivative with respect to W, that is

$$\begin{aligned} \nabla J_1^W &= WHH^T + WHH^T - XH^T - HX^T \\ &= 2(WH - X)H^T. \end{aligned} \tag{27}$$

Concatenate W^T's columns to a vector vec(W^T), then J_1^W can be written as

$$J_1^W = \text{vec}(W^T)^T \begin{bmatrix} HH^T & & \\ & \ddots & \\ & & HH^T \end{bmatrix} \text{vec}(W^T) + W\text{'s linear terms} + C_3, \tag{28}$$

where C_3 is constant. The second-order gradient of J_1^W with respect to vec(W^T) is the diagonal matrix in (28) which diagonal elements are HH^T.

As J_2^W is a linear function of W, its second order gradient is zero, and the first gradient is

$$\nabla J_2^W = \frac{\partial J_2^W}{\partial W} = \mathbf{1}_m \mathbf{1}_p^T, \tag{29}$$

where $\mathbf{1}_p = [1, 1, \cdots, 1]^T$ is an p-dimensional vector.

In order to get the first- and second-order gradients of J_3^W, (21) can be rewritten as

$$\begin{aligned} \text{tr}\left(\sum_{i,j} \tilde{x}_i L_{ij} (\tilde{x}_j)^T\right) &= \text{tr}\left(\sum_{i,j} W^T x_i L_{ij} x_j^T W\right) \\ &= \text{tr}\left[W^T \left(\sum_{i,j} x_i L_{ij} x_j^T\right) W\right] \\ &= \text{tr}(W^T X L X^T W). \end{aligned} \tag{30}$$

Concatenate the columns of W to a vector vec(W), J_3^W reads

$$J_3^W = \text{vec}(W)^T \begin{bmatrix} XLX^T & & \\ & \ddots & \\ & & XLX^T \end{bmatrix} \text{vec}(W). \tag{31}$$

The second-order gradient of J_3^W with respect to vec(W^T) is the diagonal matrix in (31) which diagonal elements are XLX^T. The first gradient of J_3^W is

$$\nabla J_3^W = \frac{\partial J_3^W}{\partial W} = 2 \sum_{i,j} x_i L_{ij} (\tilde{x}_j)^T = 2 X L \tilde{X}^T. \tag{32}$$

Finally, we minimize the cost function in (19) with two similar iterative procedures. First, we use the projected gradient method as described in Section III to solve the following conditional problem.

$$\begin{aligned} W^{(t+1)} &= \arg\min f_{H^t}^{GSNMF}(W) \\ \text{s. t. } & w_{i,k} \geq 0, h_{k,j} \geq 0, \textstyle\sum_i w_{i,k} = 1, \forall k, i, j \end{aligned} \tag{33}$$

Then the same algorithm will be taken to obtain the optimal solution for the conditional problem $H^{(t+1)} = \arg\min f_{W^{t+1}}^{GSNMF}(H)$, s. t. $w_{i,k} \geq 0$, $h_{k,j} \geq 0, \forall k, i, j$.

D. Connection to Discriminant Non-negative Matrix Factorization

In this subsection, we discuss the connection between graph-preserving sparse non-negative matrix factorization and discriminant non-negative matrix factorization as proposed in [22]. The objective function of discriminant non-negative matrix factorization is

$$D_p(X||WH) = ||X - WH||_F^2 + \gamma tr[\tilde{P_w}] - \delta tr[\tilde{P_b}], \tag{34}$$

where γ and δ are positive constants. $\tilde{P}_w$ and $\tilde{P}_b$ are the within-class scatter matrix and the between-class scatter matrix, respectively. Both are defined using the projected image vectors $\tilde{x}_i = W^T x_i$. Suppose there are K classes, and the r-th class contains n_r points, where n_r is the number of samples in each pattern class. The definitions of the scatter matrices are

$$\tilde{P}_w = \sum_{r=1}^{K} \sum_{i=1}^{n_r} (\tilde{x}_i^{(r)} - \tilde{m}^{(r)})(\tilde{x}_i^{(r)} - \tilde{m}^{(r)})^T \tag{35}$$

$$\tilde{P}_b = \sum_{r=1}^{K} n_r (\tilde{m}^{(r)} - \tilde{m})(\tilde{m}^{(r)} - \tilde{m})^T, \tag{36}$$

where $\tilde{m}^{(r)}$ denotes the mean of the projected data points of class r, i.e. $\tilde{m}^{(r)} = (1/n_r) \sum_{j=1}^{n_r} \tilde{x}_j$. $\tilde{m}$ is the mean of all the projected data points, that is, $\tilde{m} = (1/n) \sum_{l=1}^{n} \tilde{x}_l$, where n is the total number of the samples, i.e. $n = \sum_{r=1}^{K} n_r$.

Now, we consider the special case where the similarity matrix is defined as

$$S = \begin{bmatrix} S_1 & & & \\ & S_2 & & 0 \\ & & \ddots & \\ 0 & & & S_K \end{bmatrix}, \text{ where } S_r = \begin{bmatrix} \frac{1}{n_r} & \frac{1}{n_r} & \cdots & \frac{1}{n_r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n_r} & \frac{1}{n_r} & \cdots & \frac{1}{n_r} \end{bmatrix} \quad r = 1, 2, \cdots, K. \tag{37}$$

That is, for any i and j, S_{ij} is $1/n_r$ when x_i and x_j both belong to the r-th class; otherwise, S_{ij} is 0. Then $D_{ii} = \sum_i S_{ij} = I$, where I is the identity matrix. The Laplacian matrix is $L = D - S = I - S$.

According to the analysis in [13], the within-class scatter matrix and between-class scatter matrix can be rewritten as

$$\tilde{P}_w = \sum_{r=1}^{K} \left[\tilde{X}_r \tilde{X}_r^T - \frac{1}{n_r} \tilde{X}_r (\mathbf{1}_r \mathbf{1}_r^T) \tilde{X}_r^T \right] = \sum_{r=1}^{K} \tilde{X}_r L_r \tilde{X}_r^T \tag{38}$$

$$\tilde{P}_b = -\tilde{X} L \tilde{X}^T + \tilde{X} \left(I - \frac{1}{n} \mathbf{1}\mathbf{1}^T \right) \tilde{X}^T = -\tilde{X} L \tilde{X}^T + C, \tag{39}$$

where $\tilde{X}_r = [\tilde{x}_1^{(r)}, \tilde{x}_2^{(r)}, \cdots, \tilde{x}_{n_r}^{(r)}]$ is a $p \times n_r$ matrix, $L_r = I - (1/n_r)(\mathbf{1}_r \mathbf{1}_r^T)$ is a $n_r \times n_r$ symmetric matrix, and $\mathbf{1}_r = [1, 1, \cdots, 1]^T$ is a n_r-dimensional column vector with all elements being one. To simplify the above equation, we utilize the following definitions: $\tilde{X} = [\tilde{X}_1, \tilde{X}_2, \cdots, \tilde{X}_K]$ is the $p \times n$ projected data matrix where each column denotes one image point. The Laplacian matrix is $L = \text{diag}(L_1, L_2, \cdots, L_K)$. Thus, we have $\tilde{P}_w = \tilde{X} L \tilde{X}^T$, where $L = I - S$. $C = I - (1/n)\mathbf{1}\mathbf{1}^T$ is a constant.

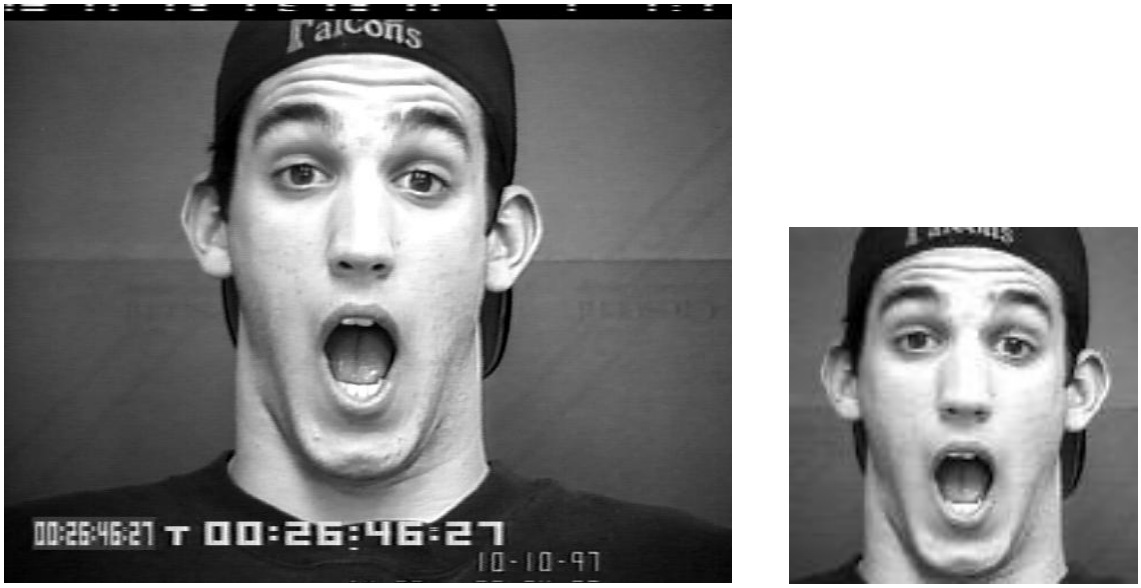
By inserting (38) and (39) to (34), the objective function of DNMF becomes

$$\begin{aligned} D_p(X||WH) &= ||X - WH||_F^2 + \gamma tr[\tilde{P}_w] - \delta tr[\tilde{P}_b] \\ &= ||X - WH||_F^2 + \gamma tr(\tilde{X} L \tilde{X}^T) - \delta tr(-\tilde{X} L \tilde{X}^T + C) \\ &= ||X - WH||_F^2 + (\gamma + \delta) tr(\tilde{X} L \tilde{X}^T) - \delta C. \end{aligned} \tag{40}$$

Thus, DNMF can be seen as a special case of the GSNMF algorithm. The discriminant constraints are exactly the graph-preserving constraints in the GSNMF algorithm if the similarity matrix is defined as in (37). For DNMF, the distribution of variation for any two points in the same class is set to be $1/n_r$. That means each sample has the same contribution to classification. While GSNMF utilizes the a penalty to make sure that the projected points retain the same local structure as the original points. The similarity matrix will incur a heavy penalty when neighboring points are mapped far apart. Furthermore, GSNMF uses a sparseness constraint whereas the standard form of DNMF does not.

V. Facial Expression Recognition Experiments

In this section, we investigate the performance of the proposed method for facial expression recognition for six basic facial expressions (namely anger, disgust, fear, happiness, sadness, and surprise). Three facial expression databases were used to verify the efficiency of the proposed algorithm. That is, Cohn-Kanade facial expression database [37], JAFFE facial expression database [38] and GENKI facial expression database [39]. Experiments were conducted on frontal facial expression images (Cohn-Kanade database and JAFFE database), partially occluded facial expression images (Cohn-Kanade database) and spontaneous facial expression images (GENKI database). In all experiments, we applied preprocessing to locate the faces. The face parts of the original facial images were cropped, and the size of each cropped image in all experiments is 60×60 pixels. Fig. 4 shows an example of the original face image and the corresponding cropped image. In this work, we use the nearest-neighbor classifier for classification. The Euclidean metric is used as the distance measure.



(a) (b)

Fig. 4. Original face image and cropped image from the Cohn-Kanade database.

A. Preprocessing

In practice, a preprocessed data matrix is used for non-negative matrix factorization. Donoho et al. [40] showed that traditional NMF cannot find the correct decomposition because all column vectors in X have a constant part. To deal with this problem, some preprocessing of the original data matrix X is necessary. The data matrix X can be written as

$$X = WH + K_0\mathbf{1}^T, \tag{41}$$

where K_0 is m-dimensional vector with elements being the minimal value of each row in X, and $\mathbf{1}$ is an n-dimensional vector with all elements equal to one. Therefore, we first subtract a constant matrix from X in order to get a more precise recovery of generating W and H. Then, the matrix $(X - K_0\mathbf{1}^T)$ is used as new data matrix to obtain the GSNMF representation of facial images.

B. Experimental Results on the Cohn-Kanade Database

The Cohn-Kanade database [37] consists of video sequences of subjects displaying distinct facial expressions, starting from a neutral expression and ending with the peak of the expression. As some

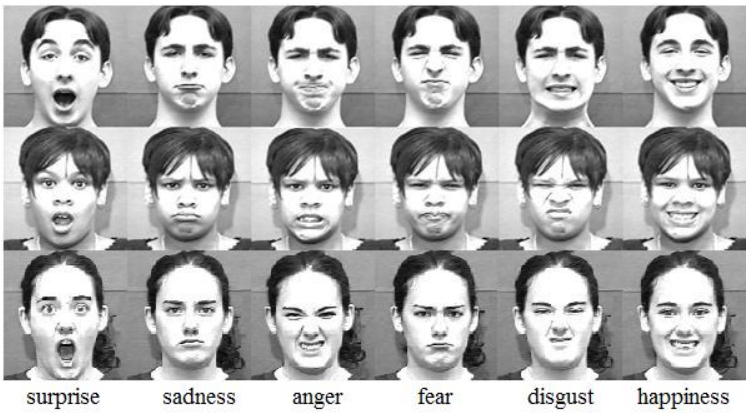


Fig. 5. Cropped face images from the Cohn-Kanade database.

subjects in the Cohn-Kanade database show less than six facial expressions, we use a subset with thirty subjects for our experiments. For each expression of a subject, the last eight frames in the videos are selected, and we treat these frames as static images for both training and testing. Some of the cropped face images in the Cohn-Kanade database with different facial expressions are shown in Fig. 5.

In the experiments, we divide the database into two subsets: training set and testing set. Experiments are conducted according to the following cases:

Case1: number of training set for one expression per person: 1; testing set: 7.

Case2: number of training set for one expression per person: 2; testing set: 6.

For each case, facial expression recognition experiments are conducted on the subset versus different dimensions (p). The tested approaches are NMF, DNMF, SNMF [41], Eigenfaces, Fisherfaces, Laplacianfaces and the proposed GSNMF. Fig. 6 shows the recognition accuracies of the two subsets versus dimensionality reduction, where Fig. 6(a) and Fig. 6(b) correspond to Case 1 and Case 2, respectively. The performance of all NMF motivated methods is reported for up to 100 basis images. As we can see, the proposed GSNMF method outperforms all other tested approaches in facial expression recognition. The best facial expression recognition accuracies obtained in Case 1 when using NMF, DNMF, SNMF, and GSNMF are 89.9%, 91.6%, 91.6% and 93.5%; the best facial expression recognition accuracies obtained in Case 2 when using NMF, DNMF, SNMF, and GSNMF are 91.3%, 92.9%, 93.0% and 94.3%. GSNMF has a recognition accuracy that is nearly 2% better than DNMF.

The parameters λ and η in the GSNMF cost function (19) should be chosen carefully. As each term in the cost function has its contribution, the final optimal solution is a tradeoff among the three terms.

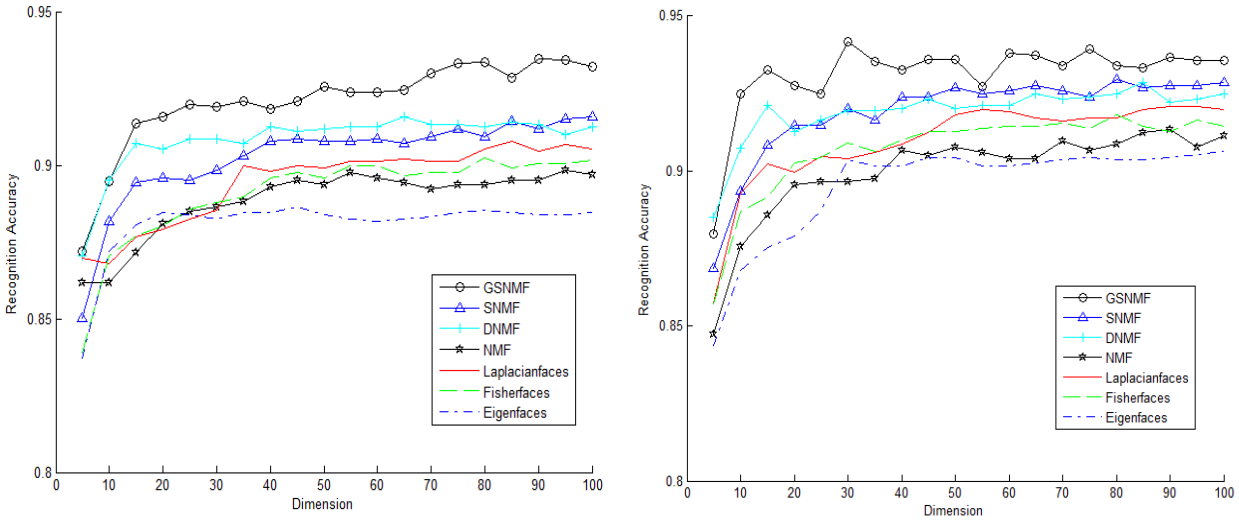


Fig. 6. Comparison of algorithms showing recognition accuracies versus dimension for the Cohn-Kanade database (a) Case 1 (b) Case 2.

In our experiments, we have tested values for λ and η by starting with small values, and changing them step by step. Good results have been obtained when choosing λ and η values in the range $[0.5, 1]$.

We also compare recognition performance with previously published on the Cohn-Kanade database. Zhou et. al. [42] utilized feature selection method to roughly classify the facial expressions and fine classification is based on rules. They obtained an average recognition accuracy of 90%. In [43], a recognition result of 90.84% using face model features and relevance vector machine was reported. Chen et. al. [44] utilized Haar-like features and Gabor features as input to weak hybrid classifiers, and achieved 93.1% average correction rate for six basic facial expressions. Our work focuses on evaluate the efficiency of GSNMF for facial feature extraction. Classifier design is not emphasized in this work. GSNMF algorithm gives comparable results by using the simplest nearest neighbor classifier, and it demonstrates the superiority of the GSNMF algorithm.

C. Experimental Results on the JAFFE Database

The JAFFE facial expression database [38] consists of 213 images of Japanese female facial expressions. Ten subjects posed 3 or 4 examples for each of the six basic expressions plus neutral face. Some of the cropped face images in the JAFFE database with different facial expressions are shown in Fig. 7. For the facial expression experiments, we randomly select two images for each expression per person to form the

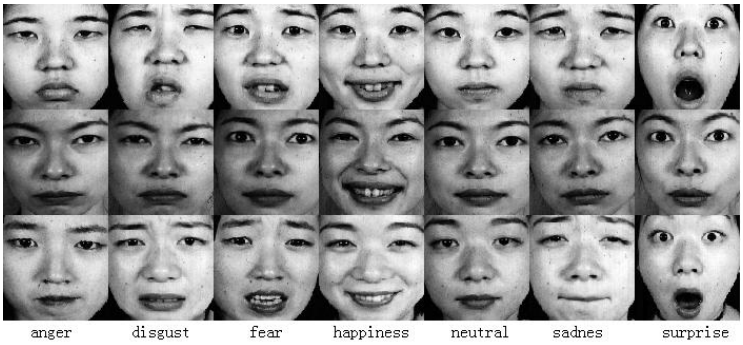


Fig. 7. Cropped face images from the JAFFE database.

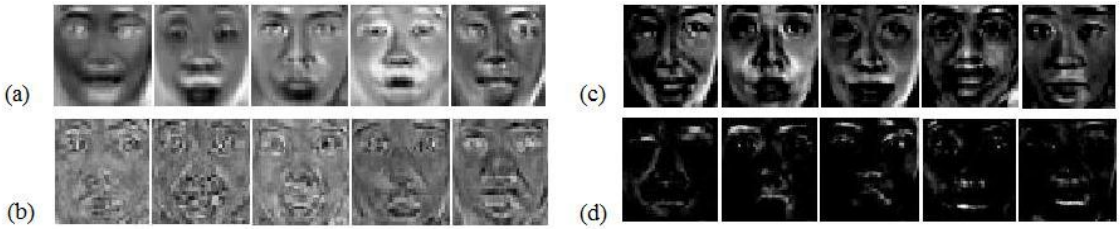


Fig. 8. Basis images extracted from the JAFFE database (a) Eigenfaces (b) Fisherfaces (c) NMFfaces (d) GSNMFfaces.

training set, and the remainder is used to test the algorithms. The facial expression recognition accuracies obtained by Eigenfaces, Fisherfaces, Laplacianfaces, NMF, DNMF, SNMF, and GSNMF are shown in Fig. 9. It can been seen that the recognition accuracies obtained for the JAFFE database are much lower than that obtained for the Cohn-Kanade database. This can be explained by the fact that the facial expressions posed in the Cohn-Kanade database are much more apparent than that in the JAFFE database. From Fig. 9, we can see that the proposed GSNMF algorithm again achieves the best recognition accuracy of the tested algorithms. GSNMF is a more efficient method for facial expression recognition.

According to the principle of non-negative matrix factorization, the face images are represented by combining multiple basis images with addition only. In contrast to PCA, no subtractions can occur. From this point of view, the basis images are expected to represent the facial parts. For comparison, five basis images corresponding to Eigenfaces, Fisherfaces, NMF, and GSNMF are shown in Fig. 8. We can see that NMF does not give an intuitive decomposition into parts. The NMF basis images reflect the holistic features of the faces rather than local features. GSNMF generates more sparse basis images which reflect distinct facial components for each facial expression.

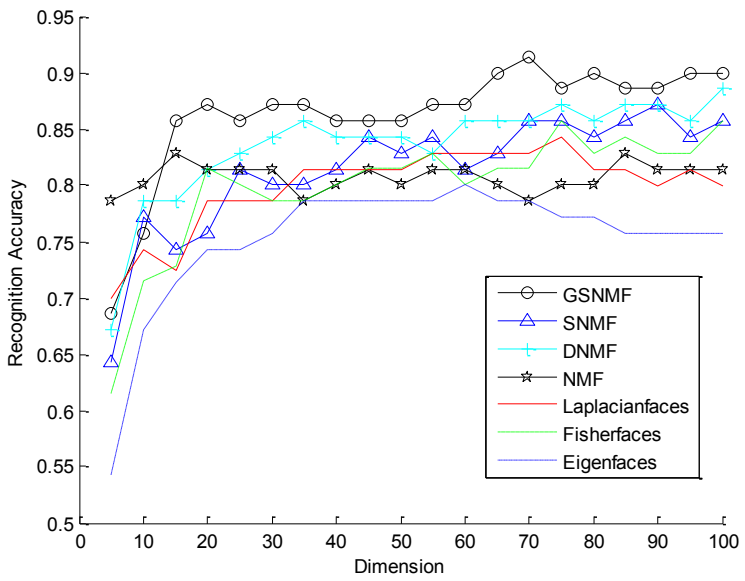


Fig. 9. Comparison of algorithms showing recognition accuracies versus dimension for the JAFFE database.

Now we compare recognition performance with previously published results. Zhang [45] used a neural network to get 90.1% recognition accuracy. In [46], a result of 92% using linear discriminant analysis was reported, but they only included nine people's face images. Guo et. al. [47] utilized selected Gabor features and compared various classifier' performances, and the linear programming feature selection with support vector machine achieved 91% recognition accuracy. In conclusion, GSNMF algorithm gives comparable results with the simplest nearest neighbor classifier.

D. Facial Expression Recognition under Partial Occlusion

GSNMF can extract parts-based facial features of the facial expression images, so it is supposed to be robust to some partial occlusion on the facial expression images. In this subsection, we verify the performance of GSNMF algorithms on facial expression images from the Cohn-Kanade database with eyes, nose, and mouth occlusions. Some preprocessing was done to get the occlusive facial expression images. An eyes mask and a nose mask, as well as a mouth mask were created to simulate partial occlusion on the facial images. Some samples from the Cohn-Kanade database under eyes, nose, and mouth region occlusion for all facial expressions are shown in Fig. 10. The experiments are conducted on occluded facial expression images in the same way as described in the previous subsection, and the randomly chosen training samples for one expression per person are fixed to 3. We compare the

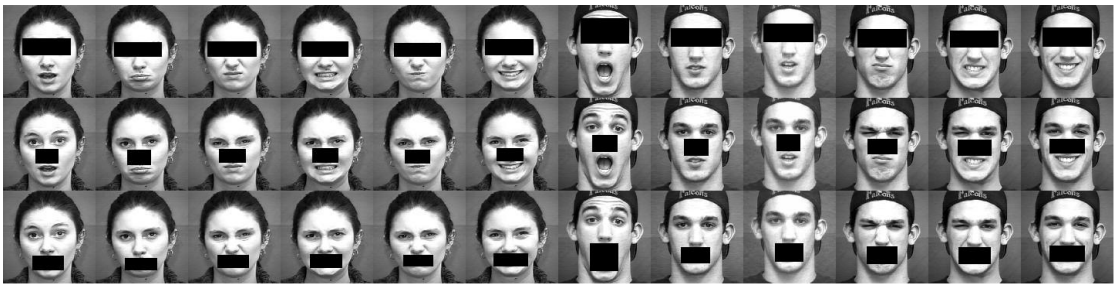


Fig. 10. Samples of facial expression images with partial occlusion from the Cohn-Kanade database.

Laplacianfaces, SNMF, and the proposed GSNMF algorithm for the occluded facial images.

The recognition accuracies of six basic facial expressions obtained by the three algorithms are shown in Fig. 11. The confusion matrix is computed for the GSNMF algorithm. The confusion matrix is a $K \times K$ matrix containing the information of the actual class (its columns) and the class label obtained through classification (its rows). The diagonal entries of the confusion matrix are the percentages corresponding to the correctly classified facial expressions, while the off-diagonal entries are the percentages corresponding to misclassification rates. The confusion matrices obtained for no occlusion, eyes occlusion, nose occlusion, and mouth occlusion are merged in Table I. The effects of the three occlusion types are analyzed as follows:

(1) Eyes occlusion

Performing facial expression recognition on facial images under eyes occlusion using GSNMF, SNMF, and Laplacianfaces, achieved 93.3%, 91.2%, and 90.0%, respectively. It seems that eyes occlusion most affects sadness, anger, surprise and disgust.

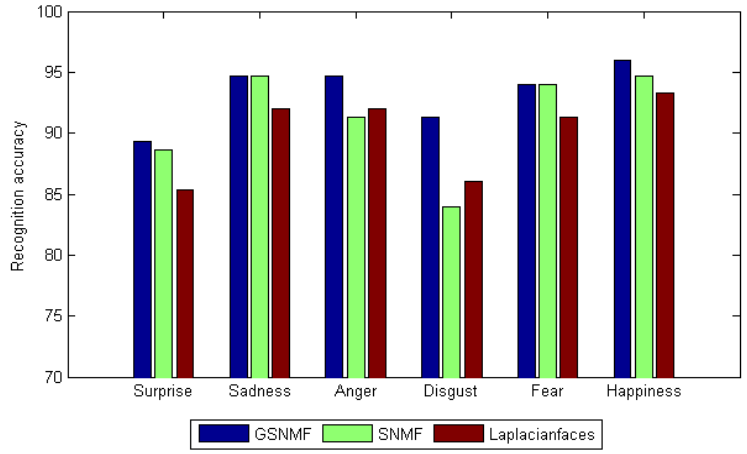
(2) Nose occlusion

Performing facial expression recognition on facial images under nose occlusion using GSNMF, SNMF, and Laplacianfaces, achieved 94.0%, 92.8%, and 91.3%, respectively. It seems that nose occlusion most affects disgust, anger, fear, and sadness.

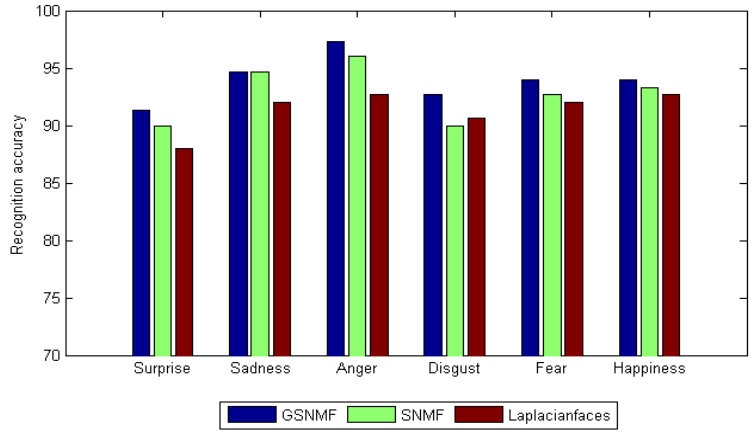
(3) Mouth occlusion

Performing facial expression recognition on facial images under mouth occlusion using GSNMF, SNMF, and Laplacianfaces, achieved 91.4%, 90.1%, and 88.8%, respectively. It seems that mouth occlusion most affects anger, fear, happiness and surprise.

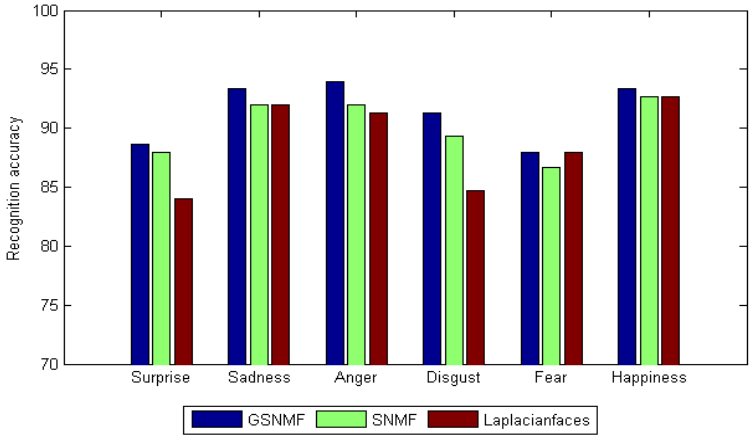
To summarize, mouth occlusion affects the facial expression recognition results the most. Therefore, the



(a)



(b)



(c)

Fig. 11. Comparison of three algorithms for six facial expressions on partially occluded facial images (a) under eyes occlusion (b) under nose occlusion (c) under mouth occlusion.

TABLE I

CONFUSION MATRIX FOR GSNMF WITH DIFFERENT OCCLUSIONS ON THE COHN-KANADE DATABASE.

		surprise	sadness	anger	disgust	fear	happiness
surprise	no	92.7	0	0	2.7	3.3	1.3
	eyes	89.3	0.7	1.3	2.7	4.0	2.0
	nose	91.3	0.7	0.7	2.7	3.3	1.3
	mouth	88.7	1.3	1.3	3.3	3.3	2.0
sadness	no	0	95.3	0	2.7	2.0	0
	eyes	0	94.7	1.3	1.3	2.7	0
	nose	0	94.7	0.7	2.0	2.7	0
	mouth	0	93.3	0	3.3	2.0	1.3
anger	no	0	0	98.7	0	0	1.3
	eyes	0	0.67	94.7	3.3	0	1.3
	nose	0	0	97.3	1.3	0	1.3
	mouth	0	2.0	94	2.7	0	1.3
disgust	no	1.3	3.3	2.0	93.3	0	0
	eyes	1.3	2.0	2.7	91.3	2.7	0
	nose	0	3.3	2.7	92.7	0	1.3
	mouth	0	4.0	3.3	91.3	0	1.3
fear	no	0	0	1.3	2.0	95.3	1.3
	eyes	0	1.3	1.3	0	94	3.3
	nose	0	0	1.3	2.0	94	2.7
	mouth	2.0	1.3	2.0	2.7	88	4.0
happiness	no	0	0	2.0	1.3	0	96.7
	eyes	0	0	2.7	1.3	0	96
	nose	0	1.3	3.3	1.3	0	94
	mouth	0	0	4.7	1.3	0.7	93.3

mouth region is the most important region for recognizing the six basic facial expressions. The region of eyes is the second important region for facial expression recognition, while nose occlusions affect mostly facial expressions which have changes in the nosewing area.

E. Spontaneous Facial Expression Recognition

Until now, most of the facial expression recognition techniques focus on optimizing performance on posed facial expressions that are collected under tightly controlled laboratory conditions on a small number of human subjects. This is mainly because authentic affective expressions are difficult to collect,

Fig. 12. Automatically detected face images from the GENKI-4K database.

and manual labeling of spontaneous facial expressions for ground truth is very time consuming, error prone and expensive [4].

However, increasing evidence in psychology and neuroscience shows that spontaneously and deliberately displayed facial behavior has differences both in utilized facial muscles and their dynamics [48]. For instance, many types of spontaneous smiles are smaller in amplitude, longer in total duration, and slower in onset and offset time than posed smile [48], [49], [50].

In this section, we verify our proposed GSNMF algorithm on spontaneous facial expressions, and the MPLab GENKI-4K Database was used. The MPLab GENKI-4K Database [39] is a subset of GENKI database, it contains 4000 face images labeled as either ”smile” or ”non-smile”, of approximately as many different human subjects. The facial images are downloaded from publicly available Internet repositories of personal web pages, and spanning a wide range of illumination conditions, head poses and personal identity.

First, we used the automatic face detection system [51] developed by Mikael Nilsson to obtain cropped face images. A correct detection accuracy of 96.3% was achieved. 3853 face images were successfully detected, containing 2114 smiles and 1739 non-smiles. In the experiments, 1000 face images (500 smiles and 500 non-smiles) were randomly selected for training and the remainder was used for testing, and the human subjects in testing set are different from that of training set. All images were first converted to gray scale. The automatically located face images were rescaled to pixels, and gray level was normalized. Some of the automatically detected face images are demonstrated in Fig. 12. In the experiments, we compare the Laplacianfaces, SNMF, and GSNMF algorithms for spontaneous facial expression images. The recognition accuracies obtained by different algorithms versus dimensions are shown in Fig. 13, and the corresponding confusion matrices are demonstrated in Table II. The proposed GSNMF algorithm achieves reasonable results on spontaneous facial expression images, even there is not so much preprocessing on the facial

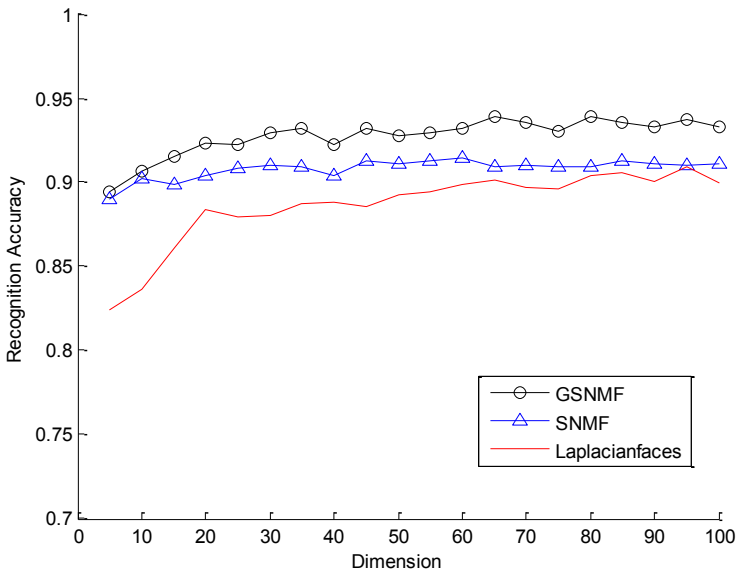


Fig. 13. Comparison of three algorithms on spontaneous facial expression images from the GENKI-4K database.

TABLE II

CONFUSION MATRICES FOR GSNMF, SNMF, AND LAPLACIANFACES ON THE GENKI-4K DATABASE (A) GSNMF (B) SNMF (C) LAPLACIANFACES.

(a)

	smile	nonsmile
smile	94.6	5.5
nonsmile	8.3	91.7

(b)

	smile	nonsmile
smile	91.8	8.2
nonsmile	9.9	90.2

(c)

	smile	nonsmile
smile	89.3	10.7
nonsmile	9.4	90.6

images. It indicates that GSNMF is efficient for facial expression representation, and its local facial representation has some tolerance for head movement.

VI. CONCLUSIONS

By considering sparse representations of original facial images and neighborhood preserving costs, we present a novel graph-preserving sparse non-negative matrix factorization (GSNMF) algorithm for facial feature extraction. Furthermore, we apply a projected gradient method to ensure that the limit point of the solution is stationary. We show that GSNMF is a more general method for non-negative solutions than DNMF, which can be incorporated in the GSNMF framework. Experimental results show that GSNMF outperforms other NMF-based algorithms and other well-known feature extraction

methods, like Eigenfaces, Fisherfaces, and Laplacianfaces in terms of recognition accuracy in all facial expression recognition experiments (no occlusion, partial occlusion and spontaneous emotions). GSNMF preserves the local structure of the samples and makes use of class label information, which is helpful for classification. Moreover, the GSNMF algorithm obtains parts-based representations of facial images, and is more robust than other tested methods for facial expression recognition.

APPENDIX A
THEORETICAL PROOF OF THE INEQUALITY (11)

First, we consider a function $f(x)$ and any vector d. According to Taylor expansions, $f(x+d)$ can be written as

$$f(x+d) = f(x) + \nabla f^T(x)d + \frac{1}{2}d^T\nabla^2 f(x)d. \tag{42}$$

So,

$$f(x+d) - f(x) = \nabla f^T(x)d + \frac{1}{2}d^T\nabla^2 f(x)d. \tag{43}$$

Hence, for two consecutive iterations $x^{(t+1)}$ and $x^{(t)}$, we have

$$f(x^{(t+1)}) - f(x^{(t)}) = \nabla f^T(x^{(t)})(x^{(t+1)} - x^{(t)}) + \frac{1}{2}(x^{(t+1)} - x^{(t)})^T\nabla^2 f(x^{(t)})(x^{(t+1)} - x^{(t)}). \tag{44}$$

The vector form of inequality (10) is

$$f(x^{(t+1)}) - f(x^{(t)}) \leq \sigma\nabla f^T(x^{(t)})(x^{(t+1)} - x^{(t)}). \tag{45}$$

According to (44), it can be written as

$$\begin{aligned} &\nabla f^T(x^{(t)})(x^{(t+1)} - x^{(t)}) + \tfrac{1}{2}(x^{(t+1)} - x^{(t)})^T\nabla^2 f(x^{(t)})(x^{(t+1)} - x^{(t)}) \leq \sigma\nabla f^T(x^{(t)})(x^{(t+1)} - x^{(t)}) \\ \Rightarrow \quad &(1-\sigma)\nabla f^T(x^{(t)})(x^{(t+1)} - x^{(t)}) + \tfrac{1}{2}(x^{(t+1)} - x^{(t)})^T\nabla^2 f(x^{(t)})(x^{(t+1)} - x^{(t)}) \leq 0. \end{aligned} \tag{46}$$

Take consideration of W instead of x, the matrix form of (46) is

$$(1-\sigma)\left\langle \nabla f_H(W^{(t)}), W^{(t+1)} - W^{(t)} \right\rangle + \frac{1}{2}\left\langle W^{(t+1)} - W^{(t)}, \left(W^{(t+1)} - W^{(t)}\right)\nabla^2 f_H(W^{(t+1)}) \right\rangle \leq 0. \tag{47}$$

REFERENCES

[1] M. Pantic and L. M. Rothkrantz, "Facial action recognition for facial expression analysis from static face images," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 34, no. 3, pp. 1449-1461, 2004.

[2] J. N. Bassili, "Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face," *Journal of Personality and Social Psychology*, vol. 37, pp. 2049-2059, 1979.

[3] P. Ekman and W. V. Friesen, "Unmasking the face," Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[4] Z. Zeng, M. Pantic, G. Roisman, and T. S. Huang, "A survey of affect recognition methods: audio, visual and spontaneous expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 1, pp. 39-58, 2009.

[5] A. Mehrabian, "Communication without words," *Psychology Today*, vol. 2, no. 4, pp. 53-56, 1968.

[6] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 34, no. 3, pp. 1588-1595, 2004.

[7] K. Anderson and P. W. McOwan, "A real-time automated system for the recognition of human facial expressions," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 36, no. 1, pp. 96-105, 2006.

[8] A. J. Calder, A. M. Burton, P. Miller, A. W. Young, and S. Akamatsu, "A principal component analysis of facial expression," *Vision Research*, vol. 41, pp. 1179-1208, 2001.

[9] W. S. Yambor, "Analysis of PCA-based and Fisher discriminant-based image recognition algorithms," Computer Science Technical Report, July, 2000.

[10] X. He and P. Niyoqi, "Locality preserving projections," *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, December 8-13, 2003.

[11] C. Shan, S. Gong, and P. W. McOwan, "A comprehensive empirical study on linear subspace methods for facial expression analysis," *in Proceeding of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, pp. 153-158, 2006.

[12] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherface: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 228-233, 1997.

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

[14] I. Buciu and I. Pitas, "Subspace image representation for facial expression analysis and face recognition and its relation to the human visual system," Organic Computing, in the series "Understanding Complex Systems", Springer, pp. 303-320, March, 2008.

[15] D. Field, "What is the goal of sensory coding?" *Neural Computation*, vol. 6, no. 4, pp. 559-601, 1994.

[16] J. W. Ellison and D. W. Massaro, "Featural evaluation, integration, and judgment of facial affect," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 23, no. 1, pp. 213-226, 1997.

[17] L. A. Ekman and W. B. Kleijn, "Improving quality prediction accuracy of P. 563 for noise suppression", *in Proceeding of the 11th International Workshop for Acoustic Echo and NOise Control*, Washington, USA, 2008.

[18] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788-791, 1999.

[19] I. Buciu and I. Pitas, "Application of non-negative and local non-negative matrix factorization to facial expression recognition," *in Proceeding of the IEEE International Conference on Pattern Recognition*, Cambridge, UK, pp. 228-291, 2004.

[20] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457-1469, 2004.

[21] S. Z. Li, X. W. Hou, H. J. Zhang and Q. S. Cheng, "Learning spatially localized, parts-based representation," *in Proceeding of the IEEE International Conference on Computer Vision and Pattern Recognition*, Hawaii, USA, pp. 207-212, 2001.

[22] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *EEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683-695, 2006.

[23] I. Buciu and I. Pitas, "A new sparse image representation algorithm applied to facial expression recognition," *in IEEE Workshop on Machine Learning for Signal Processing*, Sao Luis, Brazil, pp. 539-548, 2004.

[24] R. Zhi, M. Flierl, Q. Ruan, and W. B. Kleijn, "Facial expression recognition based on graph-preserving sparse non-negative matrix factorization," IEEE International Conference on Image Processing, Cairo, Egypt, November 7-11, 2009 (to appear).

[25] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, vol. 13, pp. 556-562, 2001.

[26] I. Kotsia, S. Zafeiriou, and I. Pitas, "A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Transactions on Information Forensics and Security*, vol.2, no. 3, pp. 588-594, 2007.

[27] C. J. Lin, "Projected gradient methods for non-negative matrix factorization," Technical report, Department of Computer Science, National Taiwan University, 2005.

[28] E. F. Gonzales and Y. Zhang, "Accelerating the Lee-Seung algorithm for non-negative matrix factorization," Technical report, Department of Computational and Applied Mathematics, Rice University, 2005.

[29] D. P. Bertsekas, "Nonlinear Programming," Athena Scientific, Belmont, MA 02178-9998, second edition, 1999.

[30] P. O. Hoyer, "Non-negative sparse coding," *in Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, Piscataway, New Jersey, pp. 557-565, 2002.

[31] E. Amaldi, and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, pp. 237-260, 1998.

[32] D. Donoho, "For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution," *Communications On Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797-829, 2006.

[33] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications On Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207-1223, 2006.

[34] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computing*, vol. 15, no.6, pp. 1373-1396, 2003.

[35] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no. 12, pp. 1615-1618, 2003.

[36] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. Roux, and M. Ouimet, "Out-of-sample extensions for LLE, ISPMAP, MDS, Eigenmaps, and spectral clustering," *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, December 8-13, 2004.

[37] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," *in Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, pp. 46-53, 2000.

[38] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with Gabor wavelets," *in Proceeding of the 3th IEEE Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 200-205, 1998.

[39] MPLab GENKI Database: http://mplab.ucsd.edu

[40] D. Donoho and V. Stodden, "When does non-negative matrix factorization given a correct decomposition into parts?" *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, December 8-13, 2003.

[41] P. O. Hoyer, Non-negative sparse coding, http://www.cs.helsinki.fi/u/phoyer/software.html

[42] G. Zhou, Y. Zhan, and J. Zhang, "Facial expression recognition based on selective feature extraction," *Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, vol. 2, pp. 412-417, 2006.

[43] D. Datcu, and L. J. M. Rothkrantz, "Facial expression recognition with relevance vector machines," *IEEE International Conference on Multimedia and Expo. (ICME)*, pp. 193-196, 2005.

[44] H. Chen, C. Huang, and C. Fu, "Hybrid-boost learning for multi-pose face detection and facial expression recognition," *Pattern Recognition*, vol. 41, pp. 1173-1185, 2008.

[45] Z. Zhang, "Feature-based facial expression recognition: sensitivity analysis and experiments with a multi-layer perceptron," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 6, pp. 893-911, 1999.

[46] M. J. Lyons, J. Budynek, S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357-1362, 1999.

[47] G. Guo, and C. R. Dyer, "Learning from examples in the small sample case: face expression recognition," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 35, no. 3, pp. 477-488, 2005.

[48] P. Ekman, and EL. L. Rosenberg, "What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system," 2nd edtion, Oxford University Press.

[49] J. F. Cohn, and K. L. Schmidt, "The timing of facial motion in posed and spontaneous smiles," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 2, pp. 1-12, 2004.

[50] M. F. Valstar, H. Gunes, and M. Pantic, "How to distinguish posed from spontaneous smiles using geometric features," *in Proceedings of the 9th International Conference on Multimodal Interfaces*, pp. 38-45, 2007.

[51] http://www.mathworks.com/matlabcentral/fileexchange/13701