Pattern Recognition 45 (2012) 4080–4091

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Subclass discriminant Nonnegative Matrix Factorization for facial image analysis

Symeon Nikitidis [b,a], Anastasios Tefas [b], Nikos Nikolaidis [b,a], Ioannis Pitas [b,a,*]

[a] Informatics and Telematics Institute, Center for Research and Technology, Hellas, Greece
[b] Department of Informatics, Aristotle University of Thessaloniki, Greece

ARTICLE INFO

Article history:
Received 4 October 2011
Received in revised form
21 March 2012
Accepted 26 April 2012
Available online 16 May 2012

Keywords:
Nonnegative Matrix Factorization
Subclass discriminant analysis
Multiplicative updates
Facial expression recognition
Face recognition

ABSTRACT

Nonnegative Matrix Factorization (NMF) is among the most popular subspace methods, widely used in a variety of image processing problems. Recently, a discriminant NMF method that incorporates Linear Discriminant Analysis inspired criteria has been proposed, which achieves an efficient decomposition of the provided data to its discriminant parts, thus enhancing classification performance. However, this approach possesses certain limitations, since it assumes that the underlying data distribution is unimodal, which is often unrealistic. To remedy this limitation, we regard that data inside each class have a multimodal distribution, thus forming clusters and use criteria inspired by Clustering based Discriminant Analysis. The proposed method incorporates appropriate discriminant constraints in the NMF decomposition cost function in order to address the problem of finding discriminant projections that enhance class separability in the reduced dimensional projection space, while taking into account subclass information. The developed algorithm has been applied for both facial expression and face recognition on three popular databases. Experimental results verified that it successfully identified discriminant facial parts, thus enhancing recognition performance.

1. Introduction

Nonnegative Matrix Factorization (NMF) [1], is an unsupervised matrix decomposition algorithm that requires both the data matrix being decomposed and the yielding factors to contain nonnegative elements. The nonnegativity constraint implies that the original data are reconstructed using only additive and no subtractive combinations of the yielded basic elements. This limitation distinguishes NMF from many other traditional dimensionality reduction algorithms, such as Principal Component Analysis (PCA) [2], Independent Component Analysis (ICA) [3,4] or Singular Value Decomposition (SVD) [5].

One of the most useful properties of NMF-based methods is that they usually produce a sparse representation of the decomposed data. Sparse coding corresponds to data representation using few basic elements that are spatially distributed, and ideally, nonoverlapping. However, because the sparseness achieved by the original NMF is somewhat of a side-effect rather than a goal, caused by the imposed nonnegativity constraints, different approaches have been proposed that attempt to control the degree to which the yielded representation is sparse. Towards this direction, Hoyer incorporated the notion of sparseness into the standard NMF decomposition function, so as the representation sparsity can be better controlled [6], while Li et al. [7] introduced localization constraints leading to a parts-based representation.

To interpret NMF parts-based image representation, consider the scenario where NMF operates either on facial images or on a text documents collection. In this scenario, the NMF training procedure aims to learn the parts of the decomposed data, which, for the first case, will correspond to different facial parts, while for the latter, to meaningful topics. Consequently, the identified basis elements when combined using appropriate weight factors, will reconstruct accurately the original facial images or text documents that have been decomposed. This parts-based representation property of NMF, is consistent with the psychological intuition of combining parts to form the whole for object representation in the human brain [8,9].

Recently, numerous specialized NMF-based algorithms applied in various problems in diverse fields have been proposed. These algorithms modify the NMF decomposition cost function by incorporating additional penalty terms in order to fulfill specific requirements, arising in each application domain. In [10] Projective NMF (PNMF) was introduced, which proved to generate a much sparser, compared to original NMF, and approximately orthogonal projection matrix, which reveals strong connections between PNMF and nonnegative PCA. Extensive theoretical and

[*] Corresponding author at: Department of Informatics, Aristotle University of Thessaloniki, Greece. Tel.: +30 2310996361.
E-mail addresses: nikitidis@aiia.csd.auth.gr (S. Nikitidis), tefas@aiia.csd.auth.gr (A. Tefas), nikolaid@aiia.csd.auth.gr (N. Nikolaidis), pitas@aiia.csd.auth.gr (I. Pitas).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.04.030

practical justifications of PNMF algorithm have been given in [11]. An extension of NMF that is applicable on mixed sign data has been attempted in [12], where the proposed framework relaxes the nonnegativity constraint on the bases matrix and considers entries with both positive and negative sign, while the weights matrix remains positively constrained. The efficiency of the presented framework has been investigated in various clustering problems.

Focusing on facial image analysis, numerous specialized NMF decomposition variants have been proposed for face recognition [7,13], facial identity verification [14] and facial expression recognition [17,18]. In such applications the entire facial image forms a feature vector and NMF aims to find its projections that optimize a given criterion. The resulting projections are then used in order to project unknown test facial images from the original high dimensional image space into a lower dimensional subspace, where the criterion under consideration is optimized. One limitation of NMF is that the decomposed images should be vectorized in order to perform the nonnegative decomposition. Consequently, this vectorization leads to significant information loss, since the local structure of the decomposed images is no longer available. In order to remedy this limitation, the 3D Nonnegative Tensor Factorization (NTF) has been introduced in the literature [19,20].

A supervised NMF learning method that aims at extracting discriminant facial parts is the Discriminant NMF (DNMF) algorithm proposed in [14,26]. DNMF incorporates discrimination criteria in the NMF factorization and achieves a more efficient decomposition of the data to its discriminant parts, thus enhancing separability between classes compared to the conventional NMF. However, the incorporation of Linear Discriminant Analysis (LDA) inspired criterion [22] inside DNMF has certain shortcomings. Firstly, LDA assumes that the sample vectors of each class are generated from underlying multivariate Gaussian distributions having a common covariance matrix but different class means. Secondly, since LDA assumes that each class is represented by a single compact data cluster, the problem of nonlinearly separable classes cannot be solved. However, this problem can be tackled if we consider that each class is partitioned into a set of disjoint clusters (subclasses) and perform a discriminant analysis aiming at subclass separation between those belonging to different classes. Typically, in real world applications, data usually do have a subclass structure. This is common e.g. in facial expression recognition, since there is no unique way that people express certain emotions and there are other factors such as facial pose, texture and illumination variations that lead to expression subclasses. If this fact is not properly addressed, the performance of NMF-based methods is significantly degraded [23].

In this work, we focus on developing a specialized NMF decomposition method that performs efficiently face and facial expression recognition, alleviating certain deficiencies inherent to these problems. Motivated by observations regarding facial expression sample distribution in the initial facial space [24] and also by the fact that NMF does not perform robustly on noisy datasets, we propose a novel algorithm, called Subclass Discriminant NMF (SDNMF). The proposed method addresses the general problem of finding discriminant projections that enhance class separability in the reduced dimensional space by imposing discriminant criteria that assume multimodality of the available train data.

To remedy the aforementioned limitations, we relax the assumption that each class is expected to consist of a single compact data cluster and regard that data inside each class form various subclasses, where each one is approximated by a Gaussian distribution. Consequently, we approximate the underlying distribution of each class by a mixture of Gaussians and employ criteria inspired by the Clustering based Discriminant Analysis (CDA) introduced in [24]. Moreover, we extend the NMF algorithm modifying its decomposition by embedding appropriate discriminant constraints and reformulating the cost function that drives the optimization process. This extension provides discriminant projections that are expected both to exhibit robustness in illumination changes and expression variations, and to enhance class separability in the reduced dimensional space. To perform SDNMF optimization, we develop multiplicative update rules that consider both samples class origin and clusters formation inside each class and prove their convergence using an appropriately designed auxiliary function.

In summary, the novel contributions of this paper are the following:

- Subclass based discriminant constraints are incorporated in the NMF decomposition cost function resulting in a specialized NMF-based method.
- Novel multiplicative update rules for optimizing SDNMF are proposed and their convergence is proven.
- A decomposition of a facial image into its discriminant parts using sparse representations is obtained.

The rest of the paper is organized as follows. The NMF algorithm and some of its most notable variants are reviewed in Section 2. Section 3 introduces the proposed SDNMF method, which incorporates subclass discriminant constraints in the NMF decomposition framework and also, draws the proposed multiplicative update rules. Section 4 describes the conducted experiments, verifying the efficiency of our algorithm in face and facial expression recognition. Finally, convergence proof of our optimization scheme is provided in Appendix A, whereas directions for future work and concluding remarks are drawn in Section 5.

2. Brief review of NMF and its most notable variants

In this section, we briefly present the NMF decomposition concept and review its discriminant variant DNMF. Without loosing generality, we shall assume that the decomposed data are facial images. Obviously, the techniques that will be described can be applied to any kind of nonnegative data.

2.1. NMF

The basic idea of NMF is to approximate a facial image by a linear combination of elements, the so called basis images, that correspond to facial parts. Let $\mathcal{I}$ be a facial image database comprised of L images belonging to n different classes and $\mathbf{X} \in R_{+}^{F \times L}$ is the nonnegative data matrix whose columns are F-dimensional feature vectors obtained by scanning row-wise each facial image in the database. Thus, $x_{i,j}$ is the ith element of the jth column vector $\mathbf{x}_j$. NMF considers factorizations of the form:

$$\mathbf{X} \approx \mathbf{ZH}, \tag{1}$$

where $\mathbf{Z} \in R_{+}^{F \times M}$ is a matrix containing the basis images, while matrix $\mathbf{H} \in R_{+}^{M \times L}$ contains the coefficients of the linear combinations of the basis images required to reconstruct each original facial image in the database. Thus, the jth facial image, represented by vector $\mathbf{x}_j$, can be approximated by the factorization $\mathbf{x}_j \approx \mathbf{Zh}_j$, where $\mathbf{h}_j$ denotes the jth weight column of matrix $\mathbf{H}$. Obviously, useful factorizations for real world applications appear when the linear subspace transformation projects data from the original F-dimensional space to a M-dimensional subspace with $M \ll F$.

To measure the cost of the decomposition in (1), one popular approach is to use the Kullback–Leibler (KL) divergence metric which is the most common approximation error measure for NMF factorization methods. The KL divergence between two vectors $\mathbf{x} = [x_1, \ldots, x_F]^T$ and $\mathbf{q} = [q_1, \ldots, q_F]^T$ is defined as

$$\mathrm{KL}(\mathbf{x} \| \mathbf{q}) \triangleq \sum_{i=1}^{F} \left(x_i \ln \frac{x_i}{q_i} + q_i - x_i \right). \tag{2}$$

Thus, the cost of the decomposition $D_{NMF}(\mathbf{X} \| \mathbf{ZH})$ in (1) can be measured as the sum of all KL divergences between all images in the database and their respective reconstructed versions

$$D_{NMF}(\mathbf{X} \| \mathbf{ZH}) \triangleq \sum_{j=1}^{L} \mathrm{KL}(\mathbf{x}_j \| \mathbf{Zh}_j)$$
$$= \sum_{j=1}^{L} \sum_{i=1}^{F} \left(x_{i,j} \ln \left(\frac{x_{i,j}}{\sum_{k=1}^{M} z_{i,k} h_{k,j}} \right) + \sum_{k=1}^{M} z_{i,k} h_{k,j} - x_{i,j} \right). \tag{3}$$

Thus, NMF algorithm factorizes the data matrix $\mathbf{X}$ into $\mathbf{ZH}$, by solving the following optimization problem:

$$\min_{\mathbf{Z},\mathbf{H}} \quad D_{NMF}(\mathbf{X} \| \mathbf{ZH}),$$
$$\text{subject to :} \quad z_{i,k} \geq 0, \quad h_{k,j} \geq 0, \quad \sum_i z_{i,k} = 1, \ \forall i,j,k. \tag{4}$$

Using an appropriately designed auxiliary function and the Expectation Maximization (EM) algorithm, it has been shown in [25] that the following multiplicative update rules update $h_{k,j}$ and $z_{i,k}$, yielding the desired factors, while guarantee a nonincreasing behavior of the cost function in (3). The update rule for the tth iteration for $h_{k,j}^{(t)}$ is given by

$$h_{k,j}^{(t)} = h_{k,j}^{(t-1)} \frac{\sum_i z_{i,k}^{(t-1)} \frac{x_{i,j}}{\sum_l z_{i,l}^{(t-1)} h_{l,j}^{(t-1)}}}{\sum_i z_{i,k}^{(t-1)}}, \tag{5}$$

while $z_{i,k}^{(t)}$ is updated by

$$'z_{i,k}^{(t)} = z_{i,k}^{(t-1)} \frac{\sum_j h_{k,j}^{(t)} \frac{x_{i,j}}{\sum_l z_{i,l}^{(t-1)} h_{l,j}^{(t)}}}{\sum_j h_{k,j}^{(t)}}, \tag{6}$$

and normalized as

$$z_{i,k}^{(t)} = \frac{'z_{i,k}^{(t)}}{\sum_l {'z}_{l,k}^{(t)}}. \tag{7}$$

2.2. NMF variants

In order to further enhance sparsity of the resulting basis images, Li et al. proposed the Local NMF (LNMF) algorithm [7] by including appropriate additional penalty terms in the NMF decomposition cost function. More precisely, LNMF aims at creating bases that can not be further decomposed into more components, while at the same time attempts to reduce redundant information between different bases. Moreover, the algorithm retains only these representation components that contain the most important information. LNMF cost function is formulated as

$$D_{LNMF}(\mathbf{X} \| \mathbf{ZH}) \triangleq D_{NMF}(\mathbf{X} \| \mathbf{ZH}) + \alpha_1 \sum_{i \neq j}^{M} [\mathbf{Z}^T \mathbf{Z}]_{i,j}$$
$$+ \alpha_2 \sum_{i=1}^{M} [\mathbf{Z}^T \mathbf{Z}]_{i,i} - \beta \sum_{i=1}^{M} [\mathbf{HH}^T]_{i,i}, \tag{8}$$

where α_1, α_2 and β are positive constants.

Another popular NMF variant is the Discriminant Nonnegative Matrix Factorization algorithm, which is an attempt to introduce discriminant constraints in the NMF decomposition cost function. The rationale behind DNMF is to incorporate LDA inspired criteria inside the NMF factorization thus, introducing information regarding how the various facial images are separated into different classes, aiming at finding basis images that correspond to discriminant salient facial parts such as eyes, nose, mouth, eyebrows, etc.

In order to incorporate discriminant constraints into the NMF decomposition, the well known Fisher discriminant criterion [21,27] has been exploited, given by

$$J(\boldsymbol{\Psi}) = \frac{\mathrm{tr}[\boldsymbol{\Psi}^T \mathbf{S}_b \boldsymbol{\Psi}]}{\mathrm{tr}[\boldsymbol{\Psi}^T \mathbf{S}_w \boldsymbol{\Psi}]}, \tag{9}$$

where $\mathrm{tr}[\cdot]$ is the matrix trace operator. This criterion attempts to find a transformation matrix $\boldsymbol{\Psi}$, that maximizes the ratio defined by the traces of the between-class and within-class scatter matrices $'\mathbf{S}_b = \boldsymbol{\Psi}^T \mathbf{S}_b \boldsymbol{\Psi}$ and $'\mathbf{S}_w = \boldsymbol{\Psi}^T \mathbf{S}_w \boldsymbol{\Psi}$ evaluated over the projected data. DNMF cost function incorporates the optimization of a similar discriminant criterion, minimizing the dispersion of the projected samples that belong to the same class around their corresponding mean, while at the same time, maximizing the scatter of the mean vectors of all classes around their global mean. Consequently, the DNMF algorithm minimizes the following cost function:

$$D_{DNMF}(\mathbf{X} \| \mathbf{ZH}) \triangleq D_{NMF}(\mathbf{X} \| \mathbf{ZH}) + \alpha \ \mathrm{tr}[{'\mathbf{S}}_w] - \beta \ \mathrm{tr}[{'\mathbf{S}}_b], \tag{10}$$

where α and β are positive constants.

Recently, Cai et al. [28] proposed the Graph regularized NMF (GNMF), designed for clustering problems, that encodes the data geometric structure. GNMF considers a nearest neighbor graph in order to exploit local geometrical invariance between the training samples when traversed from the initial space to the projection subspace. The developed optimization schema is formulated as

$$D_{GNMF}(\mathbf{X} \| \mathbf{ZH}) \triangleq \tfrac{1}{2} \| \mathbf{X} - \mathbf{ZH} \|_F^2 + \lambda \ \mathrm{tr}[\mathbf{HLH}^T], \tag{11}$$

where $\| \cdot \|_F$ is the Frobenius norm, $\lambda \geq 0$ is a regularization parameter that controls the smoothness of the new representation and $\mathbf{L}$ is the graph Laplacian matrix.

Another notable variant of NMF which exploits the training samples manifold structure in face space, is the topology preserving NMF (TPNMF) algorithm proposed by Zhang et al. in [29]. In particular, TPNMF method is specialized for face representation and recognition and achieved better discrimination, compared against the original NMF, by minimizing the constrained gradient distance which preserves the local nonlinear topology structure of the face. The developed optimization problem is formulated as

$$D_{TPNMF}(\mathbf{X} \| \mathbf{ZH}) \triangleq \tfrac{1}{2} \| \mathbf{X} - \mathbf{ZH} \|_F^2 + \lambda \left| \nabla \mathbf{ZH} \right|^2, \tag{12}$$

where $\nabla \mathbf{ZH}$ is the gradient of $\mathbf{ZH}$ and $\lambda \geq 0$ is a parameter which controls the trade-off between the minimization of the decomposition error and the topology preserving power.

3. Subclass DNMF

In this section, we present the imposed clustering based discriminant criteria and demonstrate how these are incorporated in the NMF cost function, thus creating the proposed SDNMF optimization problem. Then, we derive the proposed multiplicative update rules to optimize SDNMF and demonstrate their convergence.

3.1. Clustering based Discriminant Analysis

Similar to LDA, CDA seeks to determine a transformation matrix $\boldsymbol{\Psi}$, such that when applied on the initial input data $\mathbf{X}$, the projected samples form classes that are better separated. To do so, CDA assumes that data inside each class do not form one compact cluster, but each class can be partitioned into one or more compact subclasses. Based on this assumption, CDA attempts to discriminate classes, while, at the same time, minimizes the scatter within each subclass.

In detail, CDA exploits a modified Fisher criterion so that the between and within subclass scatter matrices are evaluated considering not only the samples class labels but also their respective subclass origins. To formulate the CDA criteria in the n-class facial image database $\mathcal{I}$, let us denote the number of subclasses composing the rth class by C_r, the total number of formed subclasses in the database by C, where $C=\sum_i^n C_i$, and the number of facial images belonging to the θth subclass of the rth class by $N_{(r)(\theta)}$. Let us also define the mean vector for the θth subclass of the rth class by $\boldsymbol{\mu}^{(r)(\theta)}=[\mu_1^{(r)(\theta)},\dots,\mu_F^{(r)(\theta)}]^T$ which is evaluated from the $N_{(r)(\theta)}$ facial images, while vector $\mathbf{x}_\rho^{(r)(\theta)}=[x_{\rho,1}^{(r)(\theta)},\dots,x_{\rho,F}^{(r)(\theta)}]^T$ corresponds to the feature vector of the ρth facial image belonging to the θth subclass of the rth class. Using the above notations, we can define the within subclass scatter matrix $\mathbf{S}_w$ as

$$\mathbf{S}_w=\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\sum_{\rho=1}^{N_{(r)(\theta)}}(\mathbf{x}_\rho^{(r)(\theta)}-\boldsymbol{\mu}^{(r)(\theta)})(\mathbf{x}_\rho^{(r)(\theta)}-\boldsymbol{\mu}^{(r)(\theta)})^T, \tag{13}$$

and the between subclass scatter matrix $\mathbf{S}_b$ as

$$\mathbf{S}_b=\sum_{i=1}^{n}\sum_{r,r\neq i}^{n}\sum_{j=1}^{C_i}\sum_{\theta=1}^{C_r}(\boldsymbol{\mu}^{(i)(j)}-\boldsymbol{\mu}^{(r)(\theta)})(\boldsymbol{\mu}^{(i)(j)}-\boldsymbol{\mu}^{(r)(\theta)})^T. \tag{14}$$

Since NMF projects the initial data to a lower dimensional subspace using the pseudo-inverse $\mathbf{Z}^\dagger=(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$, we desire to perform this projection in a discriminant manner and enhance class separability in the projection subspace. In order to determine the optimal projection of the facial images, we attempt to maximize a CDA inspired criterion formulated by exploiting the within and between subclass scatter matrices in the projection subspace. To do so, the within subclass scatter matrix evaluated on the projected samples is transformed with respect to its previous form as: $\boldsymbol{\Sigma}_w=(\mathbf{Z}^\dagger)^T\mathbf{S}_w\mathbf{Z}^\dagger$ while the between subclass scatter matrix as: $\boldsymbol{\Sigma}_b=(\mathbf{Z}^\dagger)^T\mathbf{S}_b\mathbf{Z}^\dagger$. Let us define the projected ρth facial image $\mathbf{x}_\rho^{(r)(\theta)}$ by the M-dimensional feature vector $\boldsymbol{\eta}_\rho^{(r)(\theta)}=[\eta_{\rho,1}^{(r)(\theta)},\dots,\eta_{\rho,M}^{(r)(\theta)}]^T$ resulting by applying the transformation $\boldsymbol{\eta}_\rho^{(r)(\theta)}=\mathbf{Z}^\dagger\mathbf{x}_\rho^{(r)(\theta)}$. Using the above notations we can evaluate the within and between subclass scatter matrices $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_b$ in the projection subspace as

$$\boldsymbol{\Sigma}_w=\sum_{r=1}^{n}\sum_{\theta=1}^{C_r}\sum_{\rho=1}^{N_{(r)(\theta)}}(\boldsymbol{\eta}_\rho^{(r)(\theta)}-\boldsymbol{\mu}^{(r)(\theta)})(\boldsymbol{\eta}_\rho^{(r)(\theta)}-\boldsymbol{\mu}^{(r)(\theta)})^T, \tag{15}$$

$$\boldsymbol{\Sigma}_b=\sum_{i=1}^{n}\sum_{r,r\neq i}^{n}\sum_{j=1}^{C_i}\sum_{\theta=1}^{C_r}(\boldsymbol{\mu}^{(i)(j)}-\boldsymbol{\mu}^{(r)(\theta)})(\boldsymbol{\mu}^{(i)(j)}-\boldsymbol{\mu}^{(r)(\theta)})^T, \tag{16}$$

where the M-dimensional mean vector $\boldsymbol{\mu}^{(i)(j)}$ henceforth denotes the mean vector evaluated over the projected samples composing the jth subclass of the ith class.

It is reasonable to desire the dispersion of those projected samples that belong to the same subclass of a certain class, to be as small as possible, since this would denote a high concentration of these samples around their subclass mean and consequently, more compact subclasses formation. Furthermore, to separate in the projection subspace subclasses belonging to different classes, we desire to maximize the difference between the means of every subclass of a certain class to every subclass of each other class. Therefore, we attempt to simultaneously minimize $\mathrm{tr}[\boldsymbol{\Sigma}_w]$ and maximize $\mathrm{tr}[\boldsymbol{\Sigma}_b]$.

3.2. Subclass discriminant Nonnegative Matrix Factorization (SDNMF)

In order to formulate the new cost function for the SDNMF problem we add appropriate penalty terms in the NMF decomposition as follows:

$$D_{SDNMF}(\mathbf{X}\|\mathbf{ZH})\triangleq D_{NMF}(\mathbf{X}\|\mathbf{ZH})+\frac{\alpha}{2}\,\mathrm{tr}[\boldsymbol{\Sigma}_w]-\frac{\beta}{2}\,\mathrm{tr}[\boldsymbol{\Sigma}_b], \tag{17}$$

where α and β are positive constants, while $\frac{1}{2}$ is used to simplify subsequent derivations. Consequently, the new minimization problem is formulated as

$$\min_{\mathbf{Z},\mathbf{H}}\quad D_{SDNMF}(\mathbf{X}\|\mathbf{ZH}),$$
$$\text{subject to}:\quad z_{i,k}\geq 0,\quad h_{k,j}\geq 0,\quad \sum_i z_{i,k}=1,\ \forall i,j,k, \tag{18}$$

which requires the minimization of (17) subject to the nonnegativity constraints applied on the elements of both the weights matrix $\mathbf{H}$ and the basis images matrix $\mathbf{Z}$.

In order to solve the constrained optimization problem in (18), we follow a similar approach to that in [25]. It should be noted that, as in every NMF-based optimization problem, the objective function in (17) is convex either in $\mathbf{Z}$ or in $\mathbf{H}$, but nonconvex in both variables. Therefore, the iterative optimization process of the SDNMF algorithm reaches a local minimum. To do so, the proposed process successively optimizes either variable $\mathbf{Z}$ or $\mathbf{H}$ keeping the other fixed. Since the added discriminant factors in the SDNMF cost function are totally independent from the basis images matrix $\mathbf{Z}$, keeping $\mathbf{H}$ fixed and optimizing for $\mathbf{Z}$ results in the same optimization problem to that solved by the original NMF algorithm in [25] and, consequently, leads to exactly the same update formulae. Thus, we can recall the convergence proof of conventional NMF to show that (17) is nonincreasing under the update rules in (6). The interested reader is referred to [25] for more details. Moreover, in order to optimize for $\mathbf{H}$, we define an appropriate convex auxiliary function that bounds from above the objective function defining the factorization cost. If $\mathcal{G}$ is such an auxiliary function, then (17) is nonincreasing, under the update $\mathbf{H}^{(t)}=\arg\min_{\mathbf{H}}\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)})$. By iterating this update rule, we obtain a series of minimizers $\mathbf{H}^{(t)}$ that improve the objective function. Thus, we come up with the following update rule for the weight coefficients $h_{k,l}$ which for the tth iteration is defined as

$$h_{k,l}^{(t)}=\frac{A+\sqrt{A^2+4\left(\alpha-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\frac{1}{N_{(r)(\theta)}}\right)h_{k,l}^{(t-1)}\sum_i z_{i,k}^{(t-1)}\frac{x_{i,l}}{\sum_n z_{i,n}^{(t-1)}h_{n,l}^{(t-1)}}}}{2\left(\alpha-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\frac{1}{N_{(r)(\theta)}}\right)}, \tag{19}$$

where $h_{k,l}$ can be also considered as the kth feature element, in the projection subspace, of the lth image belonging to the θth cluster of the rth facial class and parameter A is defined as:

$$A=\left(\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right)\frac{1}{N_{(r)(\theta)}}\sum_{\lambda,\lambda\neq l}h_{k,\lambda}^{(t-1)}-\frac{\beta}{N_{(r)(\theta)}}\sum_{m,m\neq r}^{n}\sum_{g=1}^{C_m}\mu_k^{(m)(g)}-1. \tag{20}$$

Details regarding how the proposed update rules and the related parameter A are derived, along with a proof that the objective function is guaranteed to have a nonincreasing behavior under the updates in (19), can be found in Appendix A.

After we obtain the optimal factors $\mathbf{Z}$,$\mathbf{H}$, SDNMF necessitates to use the pseudo-inverse $\mathbf{Z}^\dagger=(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$ of the basis images matrix $\mathbf{Z}$, in order to extract the discriminant features and compute the projection to the lower dimensional feature space for an unknown test sample $\mathbf{x}_j$ as: $'\mathbf{x}_j = \mathbf{Z}^{\dagger}\mathbf{x}_j$. However, as it has been shown in [17], $\mathbf{Z}^T$ can be also used as an appropriate alternative for this purpose, since the calculation of $\mathbf{Z}^{\dagger}$ is computationally intense and may also suffer from numerical instability. Finally, we normalize $\mathbf{Z}$ after each optimization iteration using (7). We can successively update $\mathbf{Z}$ and $\mathbf{H}$ either until no significant improvement is observed on the objective function or until a predefined maximum number of iterations is reached. The iterative optimization process using the SDNMF method is summarized in Algorithm 1.

Algorithm 1. Algorithm outline for the optimization of SDNMF.

```
Input: Nonnegative data matrix $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_L]$ along with the class label and cluster origin $\{y_i,c_i\}$ associated with each training facial image $\mathbf{x}_i$ $i=1,\ldots,L$.
Output: The basis images matrix $\mathbf{Z}\in R_+^{F\times M}$ and the weights matrix $\mathbf{H}\in R_+^{M\times L}$.
Initialize: $\mathbf{Z}^{(0)}$, $\mathbf{H}^{(0)}$ and $t=1$.
repeat
    Update $\mathbf{H}^{(t)}$ given $\mathbf{Z}^{(t-1)}$ using the update rule in (19).
    Update $\mathbf{Z}^{(t)}$ given $\mathbf{H}^{(t)}$ using the update rule in (6).
    Normalize $\mathbf{Z}^{(t)}$ such as each basis image sum up to one using (7).
    $t=t+1$.
until Stopping criteria are met.
```

3.3. Lagrangian formulation

An alternative way to solve the constrained optimization problem in (18) is by introducing Lagrangian multipliers $\mathbf{\Phi}=[\phi_{i,k}]\in R^{F\times M}$ and $\mathbf{\Psi}=[\psi_{k,j}]\in R^{M\times L}$ each one associated with constraints $z_{i,k}\geq 0$ and $h_{k,j}\geq 0$, respectively. Thus, the Lagrangian function $\mathcal{L}$ is formulated as:

$$\begin{aligned}\mathcal{L} &\triangleq D_{NMF}(\mathbf{X}\|\mathbf{ZH}))+\frac{\alpha}{2}\ \mathrm{tr}[\mathbf{\Sigma}_w]-\frac{\beta}{2}\ \mathrm{tr}[\mathbf{\Sigma}_b] \\ &\quad+\sum_i\sum_k\phi_{i,k}z_{i,k}+\sum_j\sum_k\psi_{k,j}h_{k,j} \\ &= D_{NMF}(\mathbf{X}\|\mathbf{ZH}))+\frac{\alpha}{2}\ \mathrm{tr}[\mathbf{\Sigma}_w]-\frac{\beta}{2}\ \mathrm{tr}[\mathbf{\Sigma}_b]+\mathrm{tr}[\boldsymbol{\phi}\mathbf{Z}^T]+\mathrm{tr}[\boldsymbol{\psi}\mathbf{H}^T].\end{aligned} \tag{21}$$

Consequently, the optimization problem in (18) is equivalent to the minimization of the Lagrangian with respect to $\mathbf{Z},\mathbf{H}$ i.e. $\arg\min_{\mathbf{Z},\mathbf{H}}\mathcal{L}$. To minimize $\mathcal{L}$, we first obtain its partial derivatives with respect to $z_{i,j}$ and $h_{i,j}$ and set them equal to zero

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial h_{i,j}} &= -\sum_k\frac{x_{k,j}z_{k,i}}{\sum_l z_{k,l}h_{l,j}}+\sum_l z_{l,i}+\psi_{i,j}+\frac{\alpha}{2}\frac{\partial\ \mathrm{tr}[\mathbf{\Sigma}_w]}{\partial h_{i,j}}-\frac{\beta}{2}\frac{\partial\ \mathrm{tr}[\mathbf{\Sigma}_b]}{\partial h_{i,j}}=0, \\ \frac{\partial\mathcal{L}}{\partial z_{i,j}} &= -\sum_l\frac{x_{i,l}h_{j,l}}{\sum_k z_{i,k}h_{k,l}}+\sum_l h_{j,l}+\phi_{i,j}+\frac{\alpha}{2}\frac{\partial\ \mathrm{tr}[\mathbf{\Sigma}_w]}{\partial z_{i,j}}-\frac{\beta}{2}\frac{\partial\ \mathrm{tr}[\mathbf{\Sigma}_b]}{\partial z_{i,j}}=0.\end{aligned} \tag{22}$$

According to KKT conditions [30] $\phi_{i,j}z_{i,j}=0$ and also $\psi_{i,j}h_{i,j}=0$. Consequently, we obtain the following equalities:

$$\begin{aligned}\left(\frac{\partial\mathcal{L}}{\partial h_{i,j}}\right)h_{i,j}=0 &\Leftrightarrow -\sum_k\frac{x_{k,j}z_{k,i}}{\sum_l z_{k,l}h_{l,j}}h_{i,j}+ \\ &\quad\sum_l z_{l,i}h_{i,j}+\alpha(h_{i,j}-\mu_i^{(r)(\theta)})h_{i,j} \\ &\quad-\frac{\beta}{N_{(r)(\theta)}}\mu_i^{(r)(\theta)}(C-C_r)h_{i,j} \\ &\quad+\frac{\beta}{N_{(r)(\theta)}}\sum_{m,m\neq r}^{n}\sum_{g=1}^{C_m}\mu_i^{(m)(g)}h_{i,j}=0,\end{aligned} \tag{23}$$

$$\left(\frac{\partial\mathcal{L}}{\partial z_{i,j}}\right)z_{i,j}=0\Leftrightarrow-\sum_l\frac{x_{i,l}h_{j,l}}{\sum_k z_{i,k}h_{k,l}}z_{i,j}+\sum_l h_{j,l}z_{i,j}=0. \tag{24}$$

Solving the quadratic function for $h_{i,j}$, resulting from Eq. (23), leads to the multiplicative update rule in (19). On the other hand, the update rule in (6) is directly derived by solving (24) for $z_{i,j}$.

3.4. Connections to DNMF method

Considering the simplest case where each class can be ideally represented in the projection subspace by a single compact cluster, the resulting within and between subclass scatter matrices of CDA are simplified. More precisely, $\mathbf{\Sigma}_w$ degenerates to the scatter matrix of sample vectors around their class mean, while $\mathbf{\Sigma}_b$ degenerates to the scatter matrix of the class mean vectors. Consequently, CDA is essentially transformed to an LDA variant, the Nonparametric Discriminant Analysis (NDA) [22], with a scale factor parameter equal to one.

Subsequently, SDNMF degenerates to a variant of the DNMF algorithm, while the multiplicative update rule in Eq. (19) is modified as follows:

$$h_{k,l}^{(t)}=\frac{A+\sqrt{A^2+4\left(\alpha-\left[\alpha+\frac{\beta}{N_r}(n-1)\right]\frac{1}{N_r}\right)h_{k,l}^{(t-1)}\sum_i z_{i,k}^{(t-1)}\frac{x_{i,l}}{\sum_n z_{i,n}^{(t-1)}h_{n,l}^{(t-1)}}}}{2\left(\alpha-\left[\alpha+\frac{\beta}{N_r}(n-1)\right]\frac{1}{N_r}\right)}, \tag{25}$$

where $A=(\alpha+(\beta/N_r)(n-1))(1/N_r)\sum_{\lambda,\lambda\neq l}h_{k,\lambda}^{(t-1)}-(\beta/N_r)\sum_{m,m\neq r}^{n}\mu_k^{(m)}-1$, N_r denotes the number of images in the rth class and $\boldsymbol{\mu}^{(m)}$ is the mean vector of the mth class in the projection subspace.

3.5. Dividing classes into subclasses

In order to perform classes division into their respective subclasses we have followed the Nearest-Neighbor (NN) based clustering algorithm presented in [31]. This algorithm first constructs a sorted sample set $\{\mathbf{x}_{r,1},\ldots,\mathbf{x}_{r,N_r}\}$ for every class r containing the N_r training sample vectors arranged as follows: samples $\mathbf{x}_{r,1}$ and $\mathbf{x}_{r,N_r}$ are the two most distant feature vectors of class r in the initial high dimensional face space, i.e. the two sample vectors $\mathbf{x}_i$, $\mathbf{x}_j$ that maximize the Euclidean distance $\|\mathbf{x}_i-\mathbf{x}_j\|_2$ where $\|\cdot\|_2$ is the L^2 norm. The rest of the samples are then ordered such as $\mathbf{x}_{r,2}$ is the closest to $\mathbf{x}_{r,1}$ sample, while $\mathbf{x}_{r,N_r-1}$ is the closest to $\mathbf{x}_{r,N_r}$ and so on. Generally, the sample ranked in the jth position inside the ordered set is the $(j-1)$th closest to $\mathbf{x}_{r,1}$ sample and at the same time the (N_r-j)th more distant to the other extremum, sample $\mathbf{x}_{r,N_r}$.

After creating such an ordered set for each class, we subsequently divide data samples belonging to the rth class into C_r subclasses by partitioning the ordered set into C_r equally sized subsets. Regarding the optimal division of each class into subclasses, various criteria have been proposed into the literature [31,32]. In [31], it has been shown that various other clustering methods can be used but they do not affect the overall performance of the subsequent classification step, significantly. This can be attributed to the fact that only first and second order statistics of each subclass are used in the optimization criteria and, thus, precise clustering is not crucial, as long as the location and dispersion of each subclass is robustly estimated. NN clustering is a good compromise between computation speed and clustering accuracy.

4. Experimental study

We compare the performance of the proposed SDNMF method, with those of the DNMF and the conventional NMF algorithm for

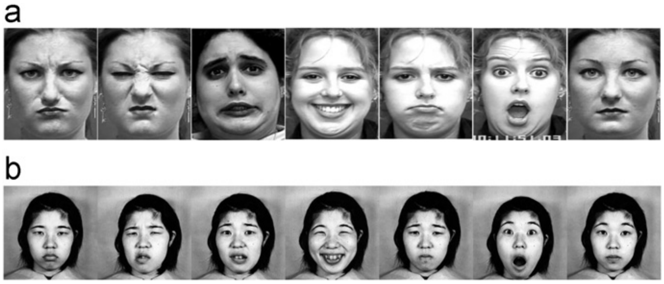

Fig. 1. Sample images depicting the different facial expressions from: (a) the CK database and (b) the JAFFE database.

facial expression recognition on two popular datasets, namely, the Cohn-Kanade (CK) [33] and the Japanese Female Facial Expression (JAFFE) [35] and for face recognition on the Extended Yale B database [16] and on the AR database [34]. Fig. 1 shows example images, from each examined facial expression dataset, depicting the seven recognized facial expressions arranged in the following order: anger, fear, disgust, happiness, sadness, surprise and neutral expression.

4.1. Preprocessing and applied experimental protocols

For the face recognition experimental comparisons we used the pre aligned and cropped versions of the Extended Yale B and AR databases, as provided by the creators, where we only anisotropically scaled the initial facial images to a fixed size of 30×40 pixels, converted to grayscale and scanned row-wise such as to form a 1200-dimensional feature vector. To measure the recognition efficacy in the Extended Yale B database we followed a similar experimental protocol as the one applied in [15]. That is, for each subject we randomly selected half of the images for training NMF, DNMF and SDNMF and learn the basis images for the low dimensional projection space, while the other half images were used for testing and report the face recognition accuracy rates. For the AR dataset we performed three different experimental scenarios of increasing difficulty which are detailed in Section 4.5.

For the facial expression recognition experiments in order to form our data collection we only acquired a single video frame from each sequence, depicting a subject performing a facial expression at its highest intensity level. Consequently, face detection was performed and the resulting facial Regions of Interest (ROIs) were manually aligned with respect to the eyes position and anisotropically scaled to a fixed size of 30×40 pixels. Finally, each such grayscale facial image was scanned row-wise, so as to form a feature vector which was used to compose either the training or the test set.

In order to measure the facial expression recognition accuracy, the following experimental protocol has been applied. We randomly partitioned the selected samples into 5-folds and a cross validation has been performed by feeding the projected discriminant facial expression representations to a linear SVM classifier [36]. This resulted into a test set formation where some expressive samples of an individual were left for testing, while his rest expressive images (depicting other facial expressions) were included in the training set. This fact significantly increased the difficulty of the expression recognition problem, since identity related issues arise. The reported average classification accuracy rate is the mean value of the percentages of the correctly classified facial expressions in all 5-folds. The value of parameters α and β was defined experimentally and the optimal values in terms of measured classification accuracy rates and convergence speed where achieved when α and β were set in the interval $(0, 1]$. Regarding the classes partitioning into subclasses we should note that since the available samples for each class in all databases are relatively few we have considered classes partitioning into 2 and 3 distinct subclasses. This partitioning is necessitated by the fact that it would be undesirable to over-segment classes and obtain subclass that contain few samples, since this will deteriorate the subsequently performed discriminant analysis. Moreover, we aim to determine for each dataset the optimal number of subclasses, that each class should be partitioned to, with respect to the reached classification performance.

4.2. Cohn-Kanade (CK) dataset

The CK AU-Coded facial expression database is among the most popular databases for benchmarking methods that perform automatic facial expression recognition. This database includes 486 image sequences depicting 97 subjects performing the 6 basic facial expressions. Each subject in each image sequence of the database poses a facial expression, starting from the neutral emotional state and finishing at the expression apex. As already mentioned, to form our data collection we neglected the intermediate video frames depicting subjects performing each facial expression in increasing intensity level and considered only the last video frame depicting each formed facial expression at its highest intensity. CK image sequences depict subjects of different racial background under severe illumination variations. Consequently, the data sample vectors do not correspond to one compact cluster per class, a fact that can be handled by the proposed SDNMF algorithm. To verify this, we considered that each facial expression class is partitioned into 3 subclasses and computed the mean expressive image for every subclass of each class. Fig. 2 shows the mean image for each facial expression considering the two more distant subclasses inside every class. Clearly the illumination variations are captured during clustering.

Fig. 3 shows the measured average facial expression recognition accuracy rates versus the projection subspace dimensionality. Moreover, Table 1 shows the confusion matrix resulted by the best performing SDNMF algorithm, where its columns and rows are named using the two letter acronyms of the respective expressions (i.e. “An”, “Di”, “Fe”, “Ha”, “Sa”, “Su” and “Ne”). The highest measured recognition accuracy rate attained by the proposed method is 69.05% while for the NMF algorithm is 64.85%. Therefore, an increase by more than 4% has been achieved by incorporating the CDA inspired discriminant constraints in the NMF cost function. As can be seen, in Fig. 3, SDNMF constantly outperforms both NMF and DNMF methods, when considering projections in a subspace of dimensionality greater than 100. However, NMF tends to achieve higher recognition accuracy rates, when the projection subspace dimensionality is less than 45. This is caused by the fact that projections to very low dimensional subspaces force both examined discriminant algorithms to extract

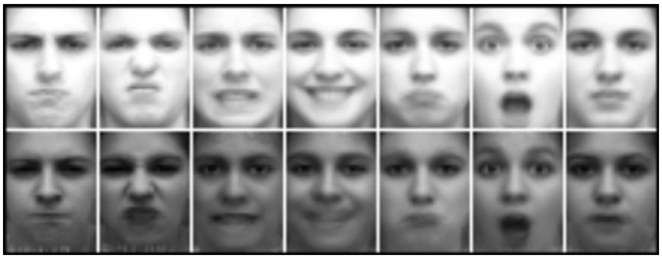

Fig. 2. Mean images for each expression considering that each facial expression class is partitioned into three clusters. Mean images are derived from the two more distant clusters inside every class. The diverse illumination conditions during facial expression capture in the CK database are evident.

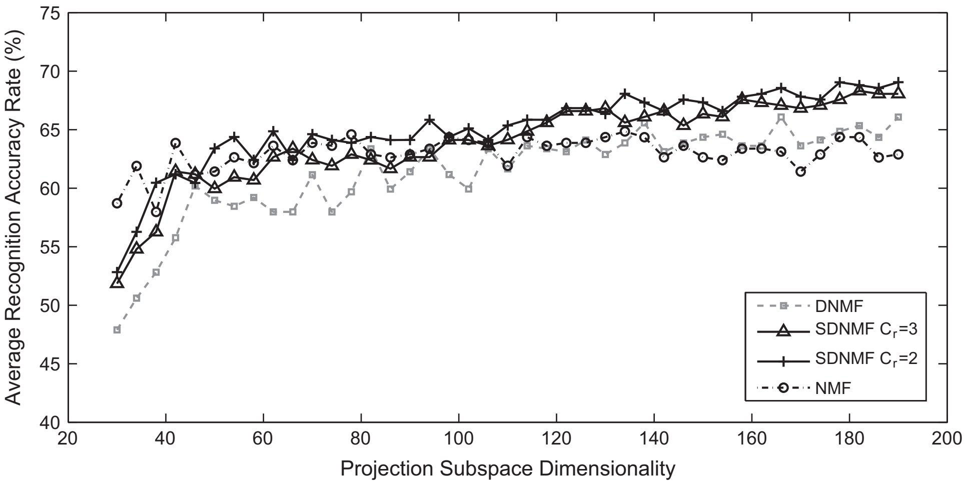

Fig. 3. Average facial expression recognition accuracy rate versus the dimensionality of the projection subspace in CK database.

Table 1
Confusion matrix for the CK database.

	An	Di	Fe	Ha	Sa	Su	Ne
An	**62.3**	8.1	3.7	0	4.5	0	7.2
Di	10.4	**68.7**	5.5	4.4	1.5	0	7.2
Fe	0	2.7	**57.2**	12.4	1.5	3.7	5.4
Ha	2.8	2.7	16.2	**77.7**	0	0	7.2
Sa	8.0	8.0	1.8	2.2	**67.8**	3.7	14.0
Su	0	0	3.7	0	6.0	**92.6**	1.8
Ne	16.5	9.8	11.9	3.3	18.7	0	**57.2**

basis images that have a holistic rather than a discriminant representation, since both algorithms try to minimize the decomposition error. This fact significantly deteriorates facial expression recognition accuracy, since holistic face representations are clearly not appropriate for facial expression discrimination. As we keep increasing the projection subspace dimensionality, SDNMF tends to generate more localized and sparse basis images that correspond to salient facial features possessing greater discrimination power.

Fig. 4 shows the yielded basis images extracted from the CK database for NMF and the proposed SDNMF algorithm considering that each class is partitioned into 2 subclasses. Both methods have been trained to find the optimal projection matrix to a subspace of equal dimensionality. As can be seen, the bases extracted by NMF have an holistic appearance and resemble distorted versions of the original facial images, while these generated by the SDNMF are sparse and many of them highlight specific local facial features such as the mouth, cheeks, eyes and eyebrows, which are salient areas in facial expression formation. This observation reveals that the proposed method successfully extracted discriminant facial features, a fact that verifies its superiority for facial expression classification tasks. Moreover, this remark is consistent with those reported in other studies proposing discriminant NMF variants [14,18,26] where it has also been found that the resulting basis images correspond to discriminant facial features.

To further investigate this remark, we examined the basis images generated by SDNMF. The upper row in Fig. 5 displays superimposed the 20 basis images with the highest associated weight for each of the seven facial expression classes, while the lower row displays those with the smallest associated weight value. As it can be observed, the first group of basis images highlight facial parts, common across all classes, that remain unaltered across any facial expression, such as the nose, the forehead and the cheeks. Consequently, these basis are assigned larger weight values, since they significantly determine the decomposition error. On the other hand, basis images ranked least according to their assigned weight value for each class, highlight characteristic facial parts unique for each facial expression, which although do not influence decisively the measured decomposition error they possess valuable information for facial expression classification. For instance, note that such basis images, in the lower row, map accurately the characteristic shape of the mouth area, during surprise and highlight the raised or lowered lip corners that are characteristic of the happiness or sadness facial expression, respectively.

4.3. JAFFE dataset

The JAFFE database contains 213 grayscale images sizing 256 × 256 pixels, depicting 10 Japanese female subjects, posing 2–4 examples of each of the 7 facial expressions. Each facial image has been labeled on these expression classes by 60 Japanese subjects. In JAFFE database NMF found to outperform SDNMF by more than 7% considering the highest achieved recognition accuracy rate for each method. We believe that this is due to the limited number of images per each facial expression class, which does not allow for a proper discriminant analysis. A similar observation is reported in [14], where DNMF was applied for face verification on a small training set thus, providing limited discriminant information to train the algorithm properly. To verify this we have created an enriched version of the JAFFE dataset using the initial facial images combined with a variety of geometrically transformed versions of each facial ROI and compared the examined algorithms on these data. To populate the training set we have considered 24 different geometrical distortions applied to each initial facial image by varying the eyes center position, during registration, by a single pixel along a cross shaped shift direction. Thus, we created in total 24 different translated, scaled and rotated versions of each original facial image in the database. Fig. 6 shows the measured recognition performance, using the enriched training set originated by the JAFFE dataset, while the test set remained unaltered. In addition, the confusion matrix corresponding to the highest performing SDNMF algorithm (i.e. considering C_r=3 and a 175-dimensional projection subspace) is shown in Table 2. The proposed method achieved the best recognition performance, while the enrichment of the training set resulted in an increase of the recognition rate

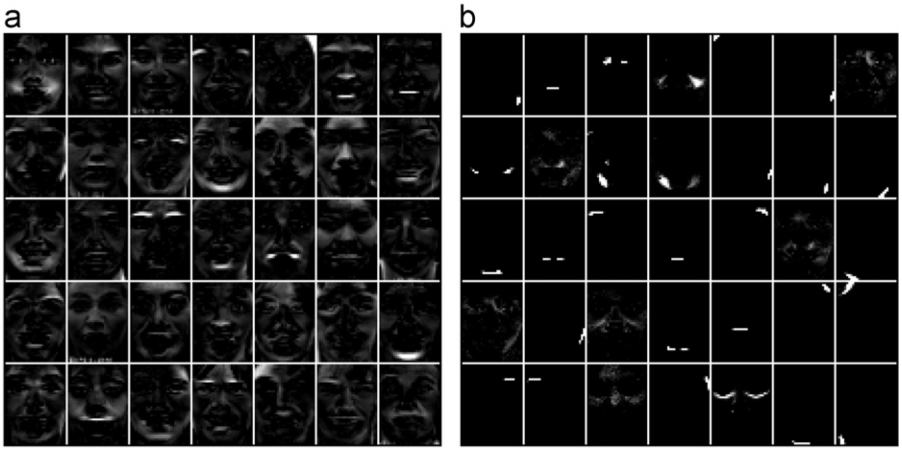

Fig. 4. Basis images yielded during training in the CK database. Basis derived from: (a) NMF and (b) SDNMF with $C_r = 2$.

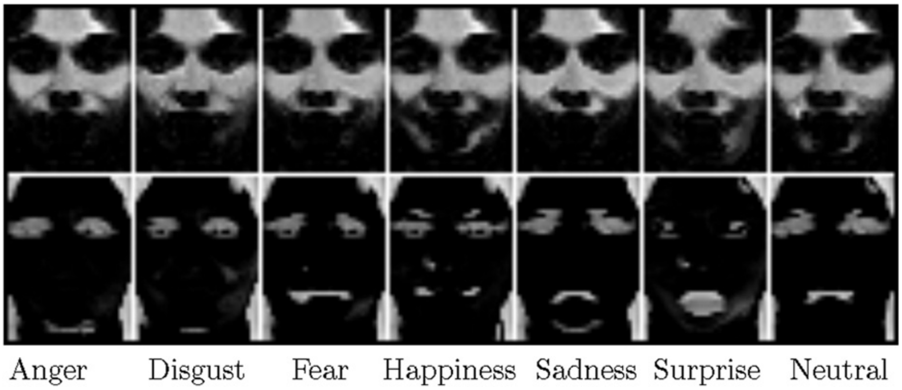

Fig. 5. Basis images, extracted from the CK database, categorized based on their corresponding weight values. The upper row shows the most common basis across each expression class, while the lower one shows the most discriminant basis images for each expression.

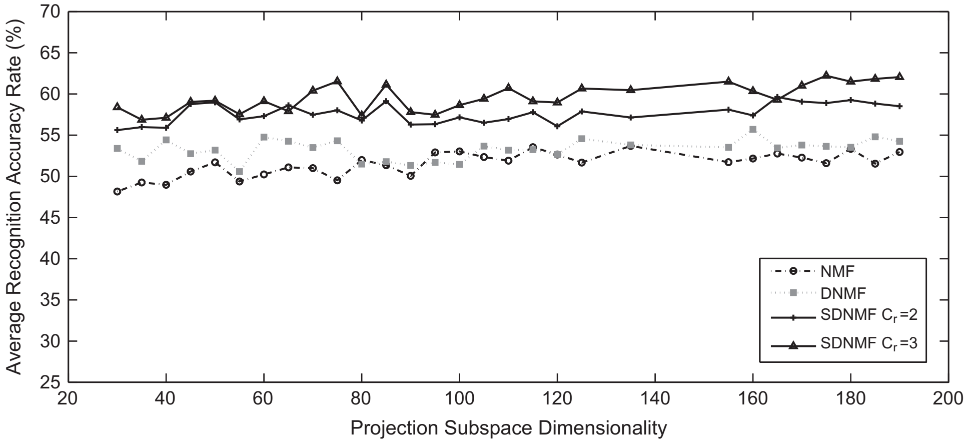

Fig. 6. Average facial expression recognition rate versus the dimensionality of the projection subspace in the enriched training set originated by the JAFFE database.

Table 2
Confusion matrix for the enriched JAFFE database.

	An	Di	Fe	Ha	Sa	Su	Ne
An	**66.7**	13.9	8.2	0	8.2	3.3	3.7
Di	13.6	**54.5**	10.1	4.2	12.3	0	3.7
Fe	0	14.2	**58.8**	9.2	8.2	0	3.7
Ha	0	2.4	2.4	**73.1**	2.1	6.7	18.9
Sa	19.7	12.6	4.2	4.2	**52.7**	6.7	10.0
Su	0	0	4.2	2.1	4.2	**76.0**	6.7
Ne	0	2.4	10.1	4.2	12.3	7.3	**53.3**

achieved by SDNMF algorithm, by almost 13% compared with that attained using the original training data.

The highest measured recognition rates achieved by each method in each examined database, as well as, the respective projection subspace dimensionality are summarized in Table 3.

4.4. Face recognition on the Extended Yale B database

Extended Yale B database consists of 2414 frontal face images of 38 individuals, captured under 64 different laboratory controlled lighting conditions. The database version used in this experimental evaluation has been manually aligned, cropped, and then resized to 168 × 192 pixels by the creators. We further reduced images size by anisotropically rescaling them to a fixed size of 30 × 40 pixels. Similarly to the experimental setup applied in [15], we have randomly selected for each subject half of the images for training, while the rest were used for testing. Searching for the optimal projection subspace, we have trained NMF, DNMF and SDNMF algorithms considering subspaces of dimensionality varying from 120 to 500. Moreover, since on average there are available 64 images for each subject, thus approximately 32 samples for each class available for training, we have considered for the SDNMF algorithm that each class is composed by either two or three disjoint subclasses.

Fig. 7 shows the attained face recognition accuracy rates of each examined method versus the projection subspace dimensionality. NMF performs consistently achieving recognition rates between 83.7% and 85.9%. On the other hand, all examined discriminant variants appear to perform worst compared to NMF, for low dimensional projection subspaces, while outperform it on subspaces of dimensionality greater than 200. SDNMF considering 2 subclasses partitioning of each class, attained the best performance among the examined methods reaching a recognition rate of 92.7%. The maximum recognition rates for NMF, DNMF and SDNMF with $C_r=3$ are 85.9%, 89.5% and 90.1%, respectively.

4.5. Face recognition on the AR dataset

AR database [34] is much more challenging than the Extended Yale B dataset and exhibits significant variations among its image samples. It contains color images corresponding to 126 different subjects depicting their frontal facial view under different facial expressions, illumination conditions, and occlusions (sunglasses and scarf). For this experiment we used the pre-aligned and cropped version of the AR database [37] containing in total 2,600 facial images of size 120 × 165 pixels corresponding to 100 different subjects captured during two sessions, separated by two weeks time. For each subject 13 images are available per each session which were scaled to a fixed size of 30 × 40 pixels and converted to grayscale.

In order to investigate our algorithms robustness we have conducted three different experiments with increasing degree of difficulty. For the first experiment, we formed our training set by considering only those facial images with illumination variations captured during the first session, while for testing we considered the respective images captured during the second recording session. For the second experiment, we used for training facial images with both varying illumination conditions and facial expressions from the first session and the respective images from the second session for testing. Finally, for the third experiment, we used all the first session images for training and the rest for testing. For all examined methods and for each applied experimental scenario we searched for the optimal projection subspace dimensionality in the interval [100,500]. Table 4 summarizes the highest attained recognition rate and the respective subspace dimensionality, by each method in each performed experiment. As it can be seen, SDNMF performs more robustly compared to both NMF and DNMF algorithms, since its efficacy does not reduce as radically across the performed different experiments. This reveals that the proposed method handles efficiently training samples variation. On the other hand, NMF and DNMF attained a decreased by 15.5% and 14.7% performance, respectively. Overall SDNMF with $C_r=2$ has been found to perform better reaching the highest recognition rate among all examined methods for all experiments.

5. Conclusion

Recent studies regarding facial expression image samples distribution in the initial face space, revealed that they do not form one compact cluster within each facial expression class, but usually images form various subclasses. This fact necessitates the

Table 3
Best average expression recognition accuracy rates in CK, JAFFE and enriched JAFFE databases.

Method	CK	JAFFE	Enriched JAFFE
SDNMF C_r=2	**69.05%** (190)	48.32% (185)	59.62%(165)
SDNMF C_r=3	68.31% (182)	49.26% (190)	**62.21%** (175)
DNMF	66.08% (166)	47.40% (178)	55.69% (160)
NMF	64.85% (134)	**56.72%** (106)	53.69% (135)

Table 4
Recognition rates for each applied experiment on the AR database.

Method	Experiment 1	Experiment 2	Experiment 3
SDNMF C_r=2	**88.5**% (500)	**85.0**% (500)	**80.1**% (500)
SDNMF C_r=3	87.3% (500)	82.1% (400)	77.6% (500)
DNMF	87.3% (500)	79.1% (500)	72.6% (500)
NMF	85.3% (500)	78.7% (300)	69.8% (400)

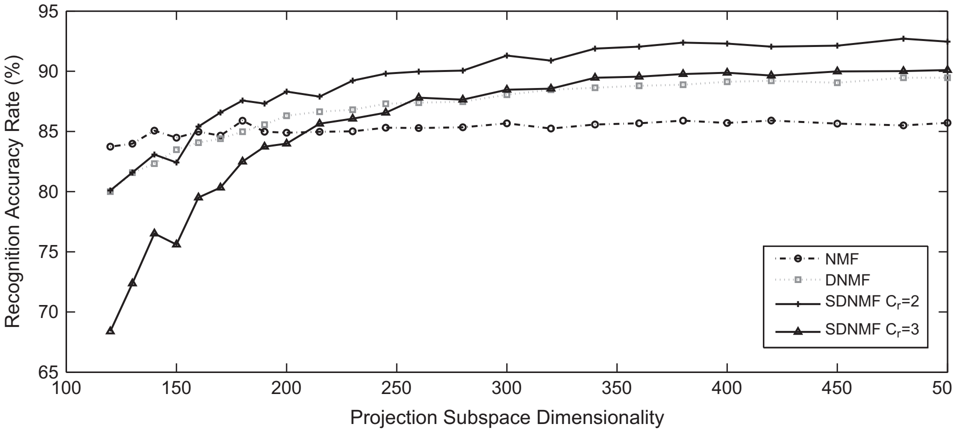

Fig. 7. Face recognition accuracy rates versus the dimensionality of the projection subspace in the Extended Yale B database.

use of CDA instead of LDA in order to form appropriate discriminant criteria. The proposed SDNMF method addresses the general problem of finding discriminant projections that enhance class separability by incorporating CDA inspired criteria in the NMF decomposition. To solve the SDNMF minimization problem, we develop novel multiplicative update rules that consider not only samples class origin but also subclasses formation inside each class and prove their convergence. We compared the performance of SDNMF with NMF and DNMF on two popular datasets for facial expression recognition and on the Extended Yale B and AR databases for face recognition. Experimental results verified the effectiveness of the proposed method on both tasks.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 211471 (i3DPost).

Appendix A. Derivation of SDNMF multiplicative update rules

Since the optimization problem is nonconvex for both variables, in order to derive the proposed update rules in (19) we generate the convex subproblem $\mathcal{O}_1(\mathbf{H})$ from (17) by fixing the basis images matrix $\mathbf{Z}$ and perform optimization for $\mathbf{H}$. Our optimization procedure makes use of an appropriately designed auxiliary function $\mathcal{G}$ that bounds the objective from above. Let $\mathcal{G}$ be an auxiliary function for $\mathcal{O}_1(\mathbf{H})$ that satisfies the following properties:

1. It bounds the objective function from above

$$\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)}) \geq \mathcal{O}_1(\mathbf{H}). \tag{A.1}$$

2. The following equation holds:

$$\mathcal{G}(\mathbf{H},\mathbf{H}) = \mathcal{O}_1(\mathbf{H}). \tag{A.2}$$

Using such an auxiliary function $\mathcal{G}$ we can derive the update rule $\mathbf{H}^{(t)} = \arg\min_{\mathbf{H}} \mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)})$ which will never increase the objective function $\mathcal{O}_1(\mathbf{H})$ since the following inequality is valid:

$$\mathcal{O}_1(\mathbf{H}^{(t)}) \leq \mathcal{G}(\mathbf{H}^{(t)},\mathbf{H}^{(t-1)}) \leq \mathcal{G}(\mathbf{H}^{(t-1)},\mathbf{H}^{(t-1)}) = \mathcal{O}_1(\mathbf{H}^{(t-1)}). \tag{A.3}$$

In order to define the auxiliary function $\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)})$, we exploit convexity of the $\ln(\cdot)$ function to derive the inequality

$$-\ln\left(\sum_k z_{i,k}h_{k,j}\right) \leq -\sum_k \alpha_k \ln\left(\frac{z_{i,k}h_{k,j}}{\alpha_k}\right), \tag{A.4}$$

which holds for all nonnegative α_k that satisfy $\sum_k \alpha_k = 1$. Setting $\alpha_k = z_{i,k}h_{k,j}^{(t-1)} / \sum_l z_{i,l}h_{l,j}^{(t-1)}$ in (A.4), we obtain the inequality

$$-\ln\left(\sum_k z_{i,k}h_{k,j}\right) \leq -\sum_k \frac{z_{i,k}h_{k,j}^{(t-1)}}{\sum_l z_{i,l}h_{l,j}^{(t-1)}} \left[\ln(z_{i,k}h_{k,j}) - \ln\left(\frac{z_{i,k}h_{k,j}^{(t-1)}}{\sum_l z_{i,l}h_{l,j}^{(t-1)}}\right)\right]. \tag{A.5}$$

From (A.5) we can derive $\mathcal{G}$ as

$$\begin{aligned}\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)}) = {} & \sum_{i,j}(x_{i,j} \ln x_{i,j} - x_{i,j}) - \sum_{i,j,k} x_{i,j} \frac{z_{i,k}h_{k,j}^{(t-1)}}{\sum_l z_{i,l}h_{l,j}^{(t-1)}} \\ & \times \left[\ln(z_{i,k}h_{k,j}) - \ln\left(\frac{z_{i,k}h_{k,j}^{(t-1)}}{\sum_l z_{i,l}h_{l,j}^{(t-1)}}\right)\right] \\ & + \sum_{i,j,k} z_{i,k}h_{k,j} + \frac{\alpha}{2}\,\mathrm{tr}[\boldsymbol{\Sigma}_w] - \frac{\beta}{2}\,\mathrm{tr}[\boldsymbol{\Sigma}_b],\end{aligned} \tag{A.6}$$

which is an appropriate auxiliary function of $\mathcal{O}_1(\mathbf{H})$ since it satisfies both desired properties. Indeed, it follows via inequality (A.5) that $\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)}) \geq \mathcal{O}_1(\mathbf{H})$ and also it is straightforward to show that $\mathcal{G}(\mathbf{H},\mathbf{H}) = \mathcal{O}_1(\mathbf{H})$.

The minimum of $\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)})$ with respect to $\mathbf{H}$ is determined by setting the gradient $\partial\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)})/\partial h_{k,l}$ equal to zero for all $h_{k,l}$. To do so the partial derivatives $\partial \mathrm{tr}[\boldsymbol{\Sigma}_w]/\partial h_{k,l}$ and $\partial\, \mathrm{tr}[\boldsymbol{\Sigma}_b]/\partial h_{k,l}$ should be evaluated. Note that the column vector $\mathbf{h}_j$ can also be considered as the projected vector to a lower dimensional feature space of the original facial vector $\mathbf{x}_j$. Assume that element $h_{k,l}$ corresponds to the kth feature of the ρth image of the θth cluster of the rth class. Thus, $h_{k,l} = \eta_{\rho,k}^{(r)(\theta)}$. Consequently, the partial derivative $\partial \mathrm{tr}[\boldsymbol{\Sigma}_w]/\partial\eta_{\rho,k}^{(r)(\theta)}$ is given by

$$\begin{aligned}\frac{\partial\, \mathrm{tr}[\boldsymbol{\Sigma}_w]}{\partial \eta_{\rho,k}^{(r)(\theta)}} &= \frac{\partial \sum_f \sum_{i=1}^{n} \sum_{j=1}^{C_i} \sum_{m=1}^{N_{(i)(j)}} (\eta_{m,f}^{(i)(j)} - \mu_f^{(i)(j)})^2}{\eta_{\rho,k}^{(r)(\theta)}} \\ &= 2\sum_{m=1}^{N_{(r)(\theta)}} (\eta_{m,k}^{(r)(\theta)} - \mu_k^{(r)(\theta)}) \left(\frac{\partial \eta_{m,k}^{(r)(\theta)}}{\partial \eta_{\rho,k}^{(r)(\theta)}} - \frac{1}{N_{(r)(\theta)}}\right) \\ &= 2\sum_{m=1}^{N_{(r)(\theta)}} (\eta_{m,k}^{(r)(\theta)} - \mu_k^{(r)(\theta)}) \frac{\partial \eta_{m,k}^{(r)(\theta)}}{\partial \eta_{\rho,k}^{(r)(\theta)}} - \frac{2}{N_{(r)(\theta)}} \sum_{m=1}^{N_{(r)(\theta)}} (\eta_{m,k}^{(r)(\theta)} - \mu_k^{(r)(\theta)}) \\ &= 2(\eta_{\rho,k}^{(r)(\theta)} - \mu_k^{(r)(\theta)}) - 2\frac{1}{N_{(r)(\theta)}} \sum_{m=1}^{N_{(r)(\theta)}} \eta_{m,k}^{(r)(\theta)} + \frac{2}{N_{(r)(\theta)}} \mu_k^{(r)(\theta)} \sum_{m=1}^{N_{(r)(\theta)}} 1 \\ &= 2(\eta_{\rho,k}^{(r)(\theta)} - \mu_k^{(r)(\theta)}) - 2\mu_k^{(r)(\theta)} + \frac{2}{N_{(r)(\theta)}} \mu_k^{(r)(\theta)} N_{(r)(\theta)} \\ &= 2(\eta_{\rho,k}^{(r)(\theta)} - \mu_k^{(r)(\theta)}).\end{aligned} \tag{A.7}$$

While the partial derivative $\partial\, \mathrm{tr}[\boldsymbol{\Sigma}_b]/\partial\eta_{\rho,k}^{(r)(\theta)}$ is computed as

$$\begin{aligned}\frac{\partial\, \mathrm{tr}[\boldsymbol{\Sigma}_b]}{\partial \eta_{\rho,k}^{(r)(\theta)}} &= \frac{\sum_f \sum_{i=1}^{n} \sum_{m,m\neq i}^{n} \sum_{j=1}^{C_i} \sum_{g=1}^{C_m} \partial(\mu_f^{(i)(j)} - \mu_f^{(m)(g)})^2}{\partial \eta_{\rho,k}^{(r)(\theta)}} \\ &= 2\sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} (\mu_k^{(r)(\theta)} - \mu_k^{(m)(g)}) \left(\frac{\partial \mu_k^{(r)(\theta)}}{\partial \eta_{\rho,k}^{(r)(\theta)}}\right) \\ &= 2\sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} (\mu_k^{(r)(\theta)} - \mu_k^{(m)(g)}) \left(\frac{1}{N_{(r)(\theta)}}\right) \\ &= 2\sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} \frac{1}{N_{(r)(\theta)}} \mu_k^{(r)(\theta)} - 2\sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} \frac{1}{N_{(r)(\theta)}} \mu_k^{(m)(g)} \\ &= \frac{2}{N_{(r)(\theta)}} \mu_k^{(r)(\theta)} \sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} 1 - \frac{2}{N_{(r)(\theta)}} \sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} \mu_k^{(m)(g)} \\ &= \frac{2}{N_{(r)(\theta)}} \mu_k^{(r)(\theta)} (C - C_r) - \frac{2}{N_{(r)(\theta)}} \sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} \mu_k^{(m)(g)}.\end{aligned} \tag{A.8}$$

Using (A.7) and (A.8) we obtain the partial derivative of $\mathcal{G}$ as

$$\begin{aligned}\frac{\mathcal{G}(\mathbf{H},\mathbf{H}^{(t-1)})}{\partial h_{k,l}} = {} & -\sum_i x_{i,l} \frac{z_{i,k}h_{k,l}^{(t-1)}}{\sum_n z_{i,n}h_{n,l}^{(t-1)}} \frac{1}{h_{k,l}} + \sum_i z_{i,k} + \alpha(h_{k,l} - \mu_k^{(r)(\theta)}) \\ & - \frac{\beta}{N_{(r)(\theta)}} (C - C_r) \mu_k^{(r)(\theta)} + \frac{\beta}{N_{(r)(\theta)}} \sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} \mu_k^{(m)(g)} = 0,\end{aligned} \tag{A.9}$$

which is a quadratic function in terms of $h_{k,l}$

$$\begin{aligned}& -\sum_i x_{i,l} \frac{z_{i,k}h_{k,l}^{(t-1)}}{\sum_n z_{i,n}h_{n,l}^{(t-1)}} + \alpha h_{k,l}^2 + \left(1 - \left[\alpha + \frac{\beta}{N_{(r)(\theta)}}(C - C_r)\right] \mu_k^{(r)(\theta)} \right. \\ & \left. + \frac{\beta}{N_{(r)(\theta)}} \sum_{m,m\neq r}^{n} \sum_{g=1}^{C_m} \mu_k^{(m)(g)} \right) h_{k,l} = 0 \Leftrightarrow\end{aligned}$$

$$\begin{aligned}
&-\sum_i x_{i,l}\frac{z_{i,k}h_{k,l}^{(t-1)}}{\sum_n z_{i,n}h_{n,l}^{(t-1)}}+\alpha h_{k,l}^2+\left(1-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\right.\\
&\frac{1}{N_{(r)(\theta)}}\left(\sum_{\lambda,\lambda\neq l}h_{k,\lambda}+h_{k,l}\right)\\
&\left.+\frac{\beta}{N_{(r)(\theta)}}\sum_{m,m\neq r}^{n}\sum_{g=1}^{C_m}\mu_k^{(m)(g)}\right)h_{k,l}=0\Leftrightarrow\\
&\left(\alpha-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\frac{1}{N_{(r)(\theta)}}\right)h_{k,l}^2\\
&+\left(1-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\frac{1}{N_{(r)(\theta)}}\sum_{\lambda,\lambda\neq l}h_{k,\lambda}\right.\\
&\left.+\frac{\beta}{N_{(r)(\theta)}}\sum_{m,m\neq r}^{n}\sum_{g=1}^{C_m}\mu_k^{(m)(g)}\right)h_{k,l}-\sum_i x_{i,l}\frac{z_{i,k}h_{k,l}^{(t-1)}}{\sum_n z_{i,n}h_{n,l}^{(t-1)}}=0.
\end{aligned}\tag{A.10}$$

Solving the quadratic equation in (A.10) leads to the proposed update rule applied in each element of matrix **H**

$$h_{k,l}^{(t)}=\frac{A+\sqrt{A^2+4\left(\alpha-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\frac{1}{N_{(r)(\theta)}}\right)h_{k,l}^{(t-1)}\sum_i z_{i,k}\frac{x_{i,l}}{\sum_n z_{i,n}h_{n,l}^{(t-1)}}}}{2\left(\alpha-\left[\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right]\frac{1}{N_{(r)(\theta)}}\right)},\tag{A.11}$$

where A is defined as

$$A=\left(\alpha+\frac{\beta}{N_{(r)(\theta)}}(C-C_r)\right)\frac{1}{N_{(r)(\theta)}}\sum_{\lambda,\lambda\neq l}h_{k,\lambda}^{(t-1)}-\frac{\beta}{N_{(r)(\theta)}}\sum_{m,m\neq r}^{n}\sum_{g=1}^{C_m}\mu_k^{(m)(g)}-1.\tag{A.12}$$

References

[1] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.

[2] I. Jolliffe, Principal Component Analysis, Springer-Verlag, New York, 1986.

[3] A. Bell, T. Sejnowski, The independent components of natural scenes are edge filters, Vision Research 37 (23) (1997) 3327–3338.

[4] P. Comon, Independent component analysis, a new concept? Signal Processing 36 (3) (1994) 287–314.

[5] G. Golub, C. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins, 1996.

[6] P. Hoyer, Non-negative matrix factorization with sparseness constraints, Journal of Machine Learning Research 5 (2004) 1457–1469.

[7] S. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2001, pp. 207–212.

[8] N. Logothetis, D. Sheinberg, Visual object recognition, Annual Review of Neuroscience 19 (1) (1996) 577–621.

[9] S. Palmer, Hierarchical structure in perceptual representation, Cognitive Psychology 9 (4) (1977) 441–474.

[10] Z. Yuan, E. Oja, Projective nonnegative matrix factorization for image compression and feature extraction, in: Proceedings of the 14th Scandinavian Conference on Image Analysis, Joensuu, Finland, 2005, pp. 333–342.

[11] Z. Yang, E. Oja, Linear and nonlinear projective nonnegative matrix factorization, IEEE Transactions on Neural Networks 21 (5) (2010) 734–749.

[12] C. Ding, T. Li, M. Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 45–55.

[13] Y. Wang, Y. Jia, C. Hu, M. Turk, Non-negative matrix factorization framework for face recognition, International Journal of Pattern Recognition and Artificial Intelligence 19 (4) (2005) 495–511.

[14] S. Zafeiriou, A. Tefas, I. Buciu, I. Pitas, Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification, IEEE Transactions on Neural Networks 17 (3) (2006) 683–695.

[15] J. Wright, A. Y Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2) (2009) 210–227.

[16] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: illumination cone models for face recognition under variable lighting and pose, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 643–660.

[17] I. Buciu, I. Pitas, A new sparse image representation algorithm applied to facial expression recognition, in: MLSP, Sao Luis, Brazil, 2004, pp. 539–548.

[18] I. Kotsia, S. Zafeiriou, I. Pitas, A novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems, IEEE Transactions on Information Forensics and Security 2 (3) (2007) 588–595.

[19] A. Shashua, T. Hazan, Non-negative tensor factorization with applications to statistics and computer vision, in: Proceedings of the 22nd International Conference on Machine Learning, ACM, 2005, pp. 792–799.

[20] S. Zafeiriou, Discriminant nonnegative tensor factorization algorithms, IEEE Transactions on Neural Networks 20 (2) (2009) 217–235.

[21] R. Fisher, The statistical utilization of multiple measurements, Annals of Eugenics 8 (1938) 376–386.

[22] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd ed., Academic Press, 1990.

[23] I. Buciu, I. Nafornita, Non-negative matrix factorization methods for face recognition under extreme lighting variations, in: IEEE International Symposium on Signals, Circuits and Systems (ISSCS), 2009, pp. 1–4.

[24] X. Chen, T. Huang, Facial expression recognition: a clustering-based approach, Pattern Recognition Letters 24 (9–10) (2003) 1295–1302.

[25] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems (NIPS), 2000, pp. 556–562.

[26] I. Buciu, I. Pitas, NMF, LNMF, and DNMF modeling of neural receptive fields involved in human facial expression perception, Journal of Visual Communication and Image Representation 17 (5) (2006) 958–969.

[27] C.R. Rao, Linear Statistical Inference and its Applications, 2nd ed., Wiley-Interscience, 2002.

[28] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized non-negative matrix factorization for data representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (8) (2011) 1548–1560.

[29] T. Zhang, B. Fang, Y. Tang, G. He, J. Wen, Topology preserving non-negative matrix factorization for face recognition, IEEE Transactions on Image Processing 17 (4) (2008) 574–584.

[30] R. Fletcher, Practical Methods of Optimization, 2nd ed., Wiley-Interscience, New York, NY, USA, 1987.

[31] M. Zhu, A. Martinez, Subclass discriminant analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (8) (2006) 1274–1286.

[32] A. Azran, Z. Ghahramani, Spectral methods for automatic multiscale data clustering, in: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2006, pp. 190–197.

[33] T. Kanade, J. Cohn, Y. Tian, Comprehensive Database for Facial Expression Analysis, 2000, pp. 46–53.

[34] A. Martinez, The AR Face Database, CVC Technical Report 24, 1998.

[35] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, in: IEEE International Conference on Automatic Face and Gesture Recognition, IEEE, 2002, pp. 200–205.

[36] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, 2001. Software available at 〈http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[37] A.M. Martinez, A.C. Kak, Pca versus lda, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2) (2001) 228–233.

Symeon Nikitidis was born in Drama, Greece in 1979. He received his diploma in 2004 from the Department of Informatics, Aristotle University of Thessaloniki and his MSc degree in 2005 from the Department of Computing Science, University of Glasgow. He is currently a researcher, teaching assistant and he is studying towards a PhD at the Artificial Intelligence and Information Analysis Laboratory at the department of Informatics, Aristotle University of Thessaloniki. His research interests include digital image/video processing, pattern recognition, face detection/recognition, as well as computer vision.

Anastasios Tefas received the BSc in informatics in 1997 and the PhD degree in informatics in 2002, both from the Aristotle University of Thessaloniki, Greece. Since 2008, he has been a lecturer at the Department of Informatics, Aristotle University of Thessaloniki. From 2006 to 2008, he was an assistant professor at the Department of Information Management, Technological Institute of Kavala. From 2003 to 2004, he was a temporary lecturer in the Department of Informatics, University of Thessaloniki. From 1997 to 2002, he was a researcher and teaching assistant in the Department of Informatics, University of Thessaloniki. He participated in 10 research projects financed by national and European funds. He has co-authored 22 journal papers, 77 papers in international conferences and contributed 7 chapters to edited books in his area of expertise. Over 1200 citations have been recorded to his publications and his H-index is 19 according to Google scholar. His current research interests include computational intelligence, pattern recognition, statistical machine learning, digital signal and image processing and computer vision.

Nikos Nikolaidis received the Diploma of Electrical Engineering in 1991 and the PhD degree in electrical engineering in 1997, both from the Aristotle University of Thessaloniki, Greece. From 1992 to 1996 he served as teaching assistant in the Departments of Electrical Engineering and Informatics at the same University. From 1998 to

2002 he was postdoctoral researcher and teaching assistant at the Department of Informatics, Aristotle University of Thessaloniki. He is currently an assistant professor in the same department. He is the co-author of the book "3-D Image Processing Algorithms" (J. Wiley, 2000). He has coauthored 6 book chapters, 22 journal papers and 82 conference papers. He currently serves as associate editor in the International Journal of Innovative Computing Information and Control and the EURASIP Journal on Image and Video Processing. His research interests include computer graphics, image and video processing and analysis, copyright protection of multimedia and 3-D image processing. He has been a scholar of the State Scholarship Foundation of Greece.

Ioannis Pitas received the Diploma of Electrical Engineering in 1980 and the PhD degree in electrical engineering in 1985 both from the Aristotle University of Thessaloniki, Greece. Since 1994, he has been a professor at the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993 he served as scientific assistant, lecturer, assistant professor, and associate professor in the Department of Electrical and Computer Engineering at the same University. He served as a visiting research associate or visiting assistant professor at several universities. He has published 153 journal papers, 400 conference papers and contributed in 22 books in his areas of interest and edited or coauthored another 5. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. In the past he served as associate editor or co-editor of four international journals and general or technical chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking and computer vision.