Fast and Orthogonal Locality Preserving Projections for Dimensionality Reduction

Rong Wang, *Member, IEEE*, Feiping Nie, Richang Hong, *Member, IEEE*, Xiaojun Chang, Xiaojun Yang, and Weizhong Yu

***Abstract*— The locality preserving projections (LPP) algorithm is a recently developed linear dimensionality reduction algorithm that has been frequently used in face recognition and other applications. However, the projection matrix in LPP is not orthogonal, thus creating difficulties for both reconstruction and other applications. As the orthogonality property is desirable, orthogonal LPP (OLPP) has been proposed so that an orthogonal projection matrix can be obtained based on a step by step procedure; however, this makes the algorithm computationally more expensive. Therefore, in this paper, we propose a fast and orthogonal version of LPP, called FOLPP, which simultaneously minimizes the locality and maximizes the globality under the orthogonal constraint. As a result, the computation burden of the proposed algorithm can be effectively alleviated compared with the OLPP algorithm. Experimental results on two face recognition data sets and two hyperspectral data sets are presented to demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*— Dimensionality reduction (DR), locality preserving projections (LPP), face recognition, hyperspectral image (HSI) classification.**

I. Introduction

IN MANY real applications of machine learning and data mining, one is often confronted with high dimensional data, such as texts, images, and videos [1]–[5]. Directly dealing with such high dimensional data is not only computationally inefficient, but also suffers from the so-called curse of dimensionality. Therefore, dimensionality reduction (DR) [6]–[12] has been proposed to represent the data in a lower dimensional space, and more importantly, to reveal the intrinsic structure of the data.

Manuscript received October 29, 2015; revised October 30, 2016 and May 14, 2017; accepted July 3, 2017. Date of publication July 12, 2017; date of current version August 7, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61401471 and in part by China Post-doctoral Science Foundation under Grant 2014M562636. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vishal Monga. *(Corresponding author: Feiping Nie.)*

R. Wang and F. Nie are with the Center for OPTical IMagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: wangrong07@tsinghua.org.cn; feipingnie@gmail.com).

R. Hong is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China.

X. Chang is with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

X. Yang is with the School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China.

W. Yu is with the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2726188

Over the past few decades, dimensionality reduction has attracted tremendous attention from large numbers of researchers, as a result of which many new algorithms have been developed. Among them, principal component analysis (PCA) [13] and linear discriminant analysis (LDA) [14] are two widely used techniques for dimensionality reduction. PCA projects the data into a lower dimensional subspace where the sample variance is maximized, while LDA extracts discriminant information by finding projection directions that maximize the ratio of the between-class and the within-class scatter. Recently, a number of studies have shown that in practical applications, high dimensional data lies in or is close to a smooth nonlinear low dimensional manifold [15]–[23]. However, both PCA and LDA only see the global Euclidean structure of data, without taking the underlying data manifold structure into consideration.

In order to uncover the manifold structure of data, dimensionality reduction approaches based on manifold learning have been developed to find a projection subspace in which the projected data is able to well preserve the intrinsic structure of data, especially the local structure. Such representative approaches include isometric feature mapping (ISOMAP) [15], Locally Linear Embedding (LLE) [16] and Laplacian eigenmaps (LE) [17], [18]. ISOMAP is an isometric manifold learning method that extends multidimensional scaling (MDS) by considering geodesic distances on a weighted graph instead of Euclidean distances. The geodesic distance between two data points is approximated by the shortest path on a constructed graph. LLE assumes that data points locally distribute on a linear patch of a manifold, preserving local linear coefficients that reconstruct each data point into a lower dimensional space by means of its neighbors. LE preserves the locality of the local neighborhoods by manipulations on an undirected graph, which indicates neighbor relations of pairwise data points. Yan *et al.* [19], [20] proposed a general dimensionality reduction framework called graph embedding, which can unify LLE, ISOMAP and Laplacian Eigenmap by reformulation. Though manifold learning methods can readily find the inherent nonlinear structure hidden in the input space, the mapping between the input space and the reduced space is implicit for all these algorithms, with definitions provided for the training data only. In addition, it is unclear how the mapping on the testing data might be evaluated, thus leading to the so-called out of sample problem.

The out of sample problem can be solved by applying a linearization procedure that explicitly builds a direct map between the input space and the reduced space. The representative approaches include locality preserving projections (LPP) [24], neighborhood preserving embedding (NPE) [25] and isometric projection (IsoP) [26]. LPP can be viewed as a linear version of Laplacian Eigenmaps that provides a way to determine the projection of testing data. Though LPP is a linear technique with locality preserving property, the projections learnt are not orthogonal. Orthogonal projections enjoy great advantages in practical applications and can make the data reconstruction much simpler. Recently, orthogonality has drawn a lot of attention in many learning problems, with several algorithms proposed to extract orthogonal projections for LPP. An orthogonal LPP algorithm is proposed in [27], although this algorithm only performs a locality minimizing projection without considering global information, which might be insufficient to provide discriminative power. Cai *et al.* [28] also proposed another orthogonal LPP (OLPP) algorithm by adopting a step by step procedure [29], which makes the algorithm computationally more expensive. Recently, Nie *et al.* [30] proposed another orthogonal LPP algorithm with a lighter computation burden than the algorithm mentioned above.

In this paper, we propose a fast and orthogonal LPP (FOLPP) algorithm that minimizes the locality and maximizes the globality simultaneously under the orthogonal constraint. Moreover, the computation burden of the proposed algorithm can be effectively alleviated. The rest of this paper is organized as follows. In Section II, we will provide a brief review of the LPP and OLPP algorithms. Section III introduces our FOLPP algorithm. In Section IV, we present the results of experiments on face recognition and hyperspectral image (HSI) classification to verify the effectiveness of the proposed algorithm. Finally, we conclude this paper in Section V.

II. Review of the LPP and OLPP

A. Locality Preserving Projections

LPP is a famous linear subspace learning algorithm [24] derived from Laplacian Eigenmap [18]. Given a training data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ with each training data point $\boldsymbol{x}_i \in \mathbb{R}^d$, in order to discover the corresponding $\boldsymbol{y}_i \in \mathbb{R}^m$ in the low dimensional manifold for $\boldsymbol{x}_i$, Laplacian Eigenmap is used to solve the following optimization problem:

$$\min_{\boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T=\boldsymbol{I}} \sum_{i,j=1}^{n} \boldsymbol{S}_{ij}\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 = \min_{\boldsymbol{Y}\boldsymbol{D}\boldsymbol{Y}^T=\boldsymbol{I}} \operatorname{tr}(\boldsymbol{Y}\boldsymbol{L}\boldsymbol{Y}^T), \tag{1}$$

where $\operatorname{tr}(\cdot)$ denotes the trace operator and $\boldsymbol{I}$ is an identity matrix. $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n] \in \mathbb{R}^{m\times n}$, and $\boldsymbol{S}_{ij}$ measures the similarity of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. $\boldsymbol{D}$ is a diagonal matrix with elements $\boldsymbol{D}_{ii} = \sum_{j=1}^{n} \boldsymbol{S}_{ij}$. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$ is a Laplacian matrix defined on a graph constructed by the training data points. A frequently used similarity matrix $\boldsymbol{S}$ could be defined by

$$\boldsymbol{S}_{ij} = \begin{cases} e^{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/t} & \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are neighbors} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where t is the heat kernel parameter and $\|\cdot\|$ denotes the ℓ_2-norm. In (2), the similarity $\boldsymbol{S}_{ij}$ monotonously increases with the decrease of the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Hence, $\boldsymbol{S}_{ij}$ incurs a heavy penalty if neighboring points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are mapped far apart [24]. The net effect of minimizing the objective function is locality preserving, i.e. if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are close then $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are close as well [24]. As stated in [20], for larger similarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the distance between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ should be smaller to minimize the objective function. Likewise, smaller similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ should lead to larger distances between $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ to ensure minimization [20].

The map from $\boldsymbol{x} \in \mathbb{R}^d$ to $\boldsymbol{y} \in \mathbb{R}^m$ learned by Laplacian Eigenmap is nonlinear, and the aim of LPP is to find a linear map to approximate this nonlinear map. Laplacian Eigenmap can only map known training data points, while LPP can easily map test data points by using a linear projection matrix. As LPP assumes the map from $\boldsymbol{x} \in \mathbb{R}^d$ to $\boldsymbol{y} \in \mathbb{R}^m$ is linear, let

$$\boldsymbol{Y} = \boldsymbol{W}^T\boldsymbol{X}, \tag{3}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m] \in \mathbb{R}^{d\times m}$ is a projection matrix.

By imposing the linear relationship (3) on optimization problem (1), LPP can be used to solve another optimization problem, as follows:

$$\min_{\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}=\boldsymbol{I}} \operatorname{tr}(\boldsymbol{W}^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W}), \tag{4}$$

From the optimization problem (4), we can see that the LPP algorithm not only performs a locality minimizing projection but also takes the global information into account. However, the learned projection matrix $\boldsymbol{W}$ is nonorthogonal, causing difficulties both for reconstruction and other applications.

The solution to this optimization problem is finally reduced to solve the following generalized eigen-decomposition problem:

$$\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{W} = \boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{\Lambda}, \tag{5}$$

where Λ is the eigenvalue matrix and $\boldsymbol{W}$ is the corresponding eigenvector matrix of $(\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T$. The total computational complexity of LPP is $2d^3 + 2d^2n + d^2m + dn^2$.

B. Orthogonal LPP

Recently, the orthogonal method has been under a spotlight, as orthogonality is desirable with good empirical performance. An orthogonal LPP (OLPP) algorithm is proposed in [28] to seek an orthogonal projection matrix $\boldsymbol{W}$ that maps the high dimensional data point $\boldsymbol{x} \in \mathbb{R}^d$ to the low dimensional data point $\boldsymbol{y} \in \mathbb{R}^m$.

Cai *et al.* [28] used a step by step procedure [29] to obtain $\boldsymbol{W} \in \mathbb{R}^{d\times m}$, which is composed of a set of orthogonal projections $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m\}$. After calculating the first $k-1$ projections $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{k-1}\}$, the kth projection $\boldsymbol{w}_k$ is calculated by solving the following optimization problem:

$$\boldsymbol{w}_k = \arg\min_{W_{k-1}^T\boldsymbol{w}_k=0} \frac{\boldsymbol{w}_k^T\boldsymbol{X}\boldsymbol{L}\boldsymbol{X}^T\boldsymbol{w}_k}{\boldsymbol{w}_k^T\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T\boldsymbol{w}_k}, \tag{6}$$

where $W_{k-1} = [w_1, w_2, \ldots, w_{k-1}]$. If we define $P_{k-1} = W_{k-1}^T (XDX^T)^{-1} W_{k-1}$, the orthogonal projections $\{w_1, w_2, \ldots, w_k\}$ can be computed using the following iterative process:

- Compute w_1 as the eigenvector of $(XDX^T)^{-1} XLX^T$ associated with the smallest eigenvalue. The computational complexity of this step is $2d^3 + 2d^2 n + d^2 + dn^2$.
- Compute w_k as the eigenvector of

$$M_k = \{I - (XDX^T)^{-1} W_{k-1} P_{k-1}^{-1} W_{k-1}^T\} \cdot ((XDX^T)^{-1} XLX^T$$

associated with the smallest eigenvalue of M_k. The computational complexity of the kth step is $3d^3 + 2d^2(k-1) + d^2 + 2d(k-1)^2 + (k-1)^3$.

However, the step by step procedure [29] makes this algorithm computationally more expensive. Moreover, the optimization objective with regard to W is unclear in this procedure [31], [32].

III. Fast and Orthogonal Locality Preserving Projections

In this section, we first introduce our FOLPP algorithm, then provide a detailed analysis of its locality preserving power.

A. The Proposed FOLPP Algorithm

In order to make the best use of both the low computation complexity (LPP) and orthogonality (OLPP), we introduce the fast and orthogonal LPP (FOLPP) algorithm by solving the following ratio trace optimization problem:

$$\min_{W^T W = I} \operatorname{tr}\left((W^T XDX^T W)^{-1} W^T XLX^T W\right). \tag{7}$$

To gain more discriminative power, it is desirable to minimize the locality and maximize the globality simultaneously. Furthermore, the learned projection matrix W satisfies the orthogonal constraint.

For notational simplicity, if we define $S_d = XDX^T$ and $S_l = XLX^T$, (7) can be rewritten as

$$\min_{W^T W = I} \operatorname{tr}\left((W^T S_d W)^{-1} W^T S_l W\right). \tag{8}$$

Let us consider first the following unconstrained ratio trace optimization problem:

$$\min_{W} \operatorname{tr}((W^T S_d W)^{-1} W^T S_l W), \tag{9}$$

which has the property that if $\hat{W}$ is the optimal solution, $\hat{W} Q$ is also an optimal solution, since

$$\operatorname{tr}((Q^T \hat{W}^T S_d \hat{W} Q)^{-1} Q^T \hat{W}^T S_l \hat{W} Q) = \operatorname{tr}((\hat{W}^T S_d \hat{W})^{-1} \hat{W}^T S_l \hat{W}),$$

where Q is an arbitrary $m \times m$ invertible matrix.

Theorem 1: VQ is the optimal solution to problem (9), where $V \in \mathbb{R}^{d \times m}$ is the generalized eigenvectors of S_l and S_d corresponding to the m smallest eigenvalues and Q is an arbitrary $m \times m$ invertible matrix.

Proof: Note that

$$\operatorname{tr}((W^T S_d W)^{-1} W^T S_l W) = \operatorname{tr}((W^T S_d W)^{-\frac{1}{2}} W^T S_l W (W^T S_d W)^{-\frac{1}{2}}) = \operatorname{tr}(V^T S_l V),$$

where $V = W(W^T S_d W)^{-\frac{1}{2}}$. Obviously, this satisfies $V^T S_d V = I$.

Therefore, problem (9) can be rewritten as

$$\min_{V^T S_d V = I} \operatorname{tr}(V^T S_l V). \tag{10}$$

The Lagrangian function of problem (10) is

$$\operatorname{tr}(V^T S_l V) - \operatorname{tr}(\lambda(V^T S_d V - I)). \tag{11}$$

In order to obtain the optimal solution to problem (10), we should determine an appropriate λ and V such that the constraint $V^T S_d V = I$ holds and the derivative of Eq. (11) w.r.t. V is equal to zero [30], [33]. Noting that λ is a symmetric matrix, suppose the eigen-decomposition of λ is $\lambda = U \Lambda U^T$, where Λ is the eigenvalue matrix of λ and U is the corresponding eigenvector matrix. By setting the derivative of Eq. (11) w.r.t. V to zero, we obtain

$$\begin{aligned} & S_l V - S_d V \lambda = 0 \\ \Rightarrow\ & S_l V = S_d V U \Lambda U^T \\ \Rightarrow\ & (S_d)^{-1} S_l V U = V U \Lambda. \end{aligned} \tag{12}$$

Let U be an identity matrix; accordingly, the Lagrangian coefficient $\lambda = \Lambda$, and Eq. (12) becomes

$$S_l V = S_d V \Lambda. \tag{13}$$

As S_d and S_l are symmetric, the V in Eq. (13) satisfies that $V^T S_d V$ is an identity matrix. Therefore, when the Lagrangian coefficient $\lambda = \Lambda$ and V is formed by the generalized eigenvectors of S_d and S_l as in Eq. (13), the constraint $V^T S_d V = I$ will hold and the derivative of Eq. (11) w.r.t. V is equal to zero. So the solution to problem (10) can be reduced to solving the generalized eigen-decomposition problem in Eq. (13), and the columns in V are formed by the generalized eigenvectors of S_d and S_l corresponding to the m smallest eigenvalues.

Recalling that $V = W(W^T S_d W)^{-\frac{1}{2}}$ and the property of problem (9), we conclude that VQ is also the optimal solution to problem (9), where Q is an arbitrary $m \times m$ invertible matrix. ∎

Theorem 2: Define QR decomposition of the optimal solution $\hat{W} \in \mathbb{R}^{d \times m}$ as $\hat{W} = \tilde{W} R$, where $\tilde{W} \in \mathbb{R}^{d \times m}$ is an orthonormal matrix and $R \in \mathbb{R}^{m \times m}$ is an upper triangular matrix. $\tilde{W}$ is the optimal solution to problem (8).

Proof: Let $Q = R^{-1}$, where R^{-1} is an invertible matrix. According to the property of problem (9), $\tilde{W} = \hat{W} Q = \hat{W} R^{-1}$ is also the optimal solution to problem (9). In addition, $\tilde{W}$ satisfies the orthogonal constraint $\tilde{W}^T \tilde{W} = I$; therefore, it is the optimal solution to problem (8). ∎

Define QR decomposition of VQ as $VQ = W_{FOLPP} R$. According to Theorem 2, W_{FOLPP} is the optimal solution to problem (8), which satisfies the orthogonal constraint $W_{FOLPP}^T W_{FOLPP} = I$. The total computational complexity of the FOLPP is $O(d^3)$. Compared with the LPP, FOLPP requires more than two steps to compute $\boldsymbol{VQ}$ and perform QR decomposition on $\boldsymbol{VQ}$, with computational complexity of dm^2 and $O(d^2m)$ respectively. In other words, the computational complexity of FOLPP is slightly higher than that of LPP.

The FOLPP algorithm is briefly stated below:

1) **Input**: Given a training data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$.
2) **Constructing the Adjacency Graph**: Let $\boldsymbol{G}$ denotes a graph with n nodes. The ith node corresponds to the image $\boldsymbol{x}_i$. We put an edge between nodes i and j if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are "close," i.e., $\boldsymbol{x}_i$ is among the k nearest neighbors of $\boldsymbol{x}_j$ or $\boldsymbol{x}_j$ is among the k nearest neighbors of $\boldsymbol{x}_i$. Note that, if the class label is available, we simply put an edge between two data points belonging to the same class.
3) **Choosing the Weights**: If node i and j are connected, set $\boldsymbol{S}_{ij} = e^{-\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2/t}$. Otherwise, set $\boldsymbol{S}_{ij} = 0$. The similarity matrix $\boldsymbol{S}$ of graph $\boldsymbol{G}$ models the local structure of the face manifold.
4) **Computing $\boldsymbol{V}$**: Compute the generalized eigen-decomposition problem:
$$\boldsymbol{S}_l\boldsymbol{V} = \boldsymbol{S}_d\boldsymbol{V}\boldsymbol{\Lambda},$$
where $\boldsymbol{S}_d = \boldsymbol{XDX}^T$, $\boldsymbol{S}_l = \boldsymbol{XLX}^T$, $\boldsymbol{D}_{ii} = \sum_{j=1}^{n}\boldsymbol{S}_{ij}$ and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$. The columns in $\boldsymbol{V}$ are formed by the generalized eigenvectors of $\boldsymbol{S}_l$ and $\boldsymbol{S}_d$ corresponding to the m smallest eigenvalues.
5) **Computing $\boldsymbol{W}_{FOLPP}$**: Perform QR decomposition on $\boldsymbol{VQ}$
$$\boldsymbol{W}_{FOLPP} = \mathrm{QR}\{\boldsymbol{VQ}\},$$
where $\boldsymbol{Q}$ is an arbitrary $m \times m$ invertible matrix. Note that $\boldsymbol{V}$ is the optimal solution of LPP, hence the optimal solution $\boldsymbol{W}_{FOLPP}$ is interesting. $\boldsymbol{W}_{FOLPP}$ is computed by performing QR decomposition on $\boldsymbol{VQ}$ rather than $\boldsymbol{V}$, where $\boldsymbol{VQ}$ is the column space of $\boldsymbol{V}$. Therefore, the optimal solution of FOLPP is different from that of LPP.

B. Locality Preserving Power

According to [28], the locality preserving power is evaluated by minimizing the locality preserving function
$$f(\boldsymbol{w}) = \frac{\boldsymbol{w}^T\boldsymbol{XLX}^T\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{XDX}^T\boldsymbol{w}} = \frac{\boldsymbol{w}^T\boldsymbol{S}_l\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_d\boldsymbol{w}}, \tag{14}$$
which reflects the locality preserving power of the projection vector $\boldsymbol{w}$. In the LPP algorithm, $\boldsymbol{V} \in \mathbb{R}^{d\times m}$ denotes the projection matrix and is obtained by solving the generalized eigenvalue decomposition problem:
$$\boldsymbol{S}_l\boldsymbol{V} = \boldsymbol{S}_d\boldsymbol{V}\boldsymbol{\Lambda}, \tag{15}$$
where $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m]$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$. $\boldsymbol{v}_k \in \mathbb{R}^{d\times 1}$ denotes the k-th projection vector and $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_m$. Problem (15) can be further rewritten as
$$[\boldsymbol{S}_l\boldsymbol{v}_1, \boldsymbol{S}_l\boldsymbol{v}_2, \ldots, \boldsymbol{S}_l\boldsymbol{v}_m] = [\lambda_1\boldsymbol{S}_d\boldsymbol{v}_1, \lambda_2\boldsymbol{S}_d\boldsymbol{v}_2, \ldots, \lambda_m\boldsymbol{S}_d\boldsymbol{v}_m].$$

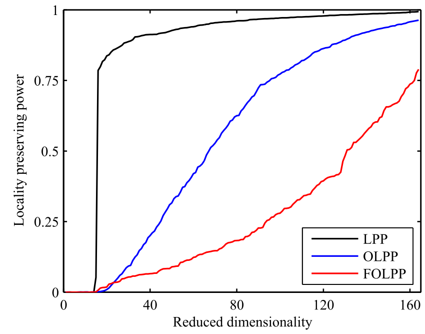


Fig. 1. Locality preserving power of the LPP, OLPP and FOLPP on the Yale data set.

That is to say, $\boldsymbol{S}_l\boldsymbol{v}_k = \lambda_k\boldsymbol{S}_d\boldsymbol{v}_k$ and the locality preserving function $f(\boldsymbol{v}_k)$ of the LPP can be computed as
$$f(\boldsymbol{v}_k) = \frac{\boldsymbol{v}_k^T\boldsymbol{S}_l\boldsymbol{v}_k}{\boldsymbol{v}_k^T\boldsymbol{S}_d\boldsymbol{v}_k} = \frac{\lambda_k\boldsymbol{v}_k^T\boldsymbol{S}_d\boldsymbol{v}_k}{\boldsymbol{v}_k^T\boldsymbol{S}_d\boldsymbol{v}_k} = \lambda_k, \tag{16}$$
which indicates that the k-th eigenvalue λ_k reflects the locality preserving power of the LPP algorithm.

In the FOLPP algorithm, the projection matrix $\boldsymbol{W}_{FOLPP}$ is obtained by performing QR decomposition on $\boldsymbol{VQ}$. Then, $\boldsymbol{VQ} = \boldsymbol{W}_{FOLPP}\boldsymbol{R}$ can be rewritten as $\boldsymbol{W}_{FOLPP} = \boldsymbol{VZ}$, where $\boldsymbol{Z} = \boldsymbol{QR}^{-1}$. Let $\boldsymbol{W}_{FOLPP} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m]$ and $\boldsymbol{Z} = [z_1, z_2, \ldots, z_m]$, such that $\boldsymbol{W}_{FOLPP} = \boldsymbol{VZ}$ can be further rewritten as
$$[\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_m] = [\boldsymbol{V}z_1, \boldsymbol{V}z_2, \ldots, \boldsymbol{V}z_m].$$
In other words, $\boldsymbol{w}_k = \boldsymbol{V}z_k$, which indicates that $\boldsymbol{w}_k$ is the linear combination of the column vectors of $\boldsymbol{V}$. The locality preserving function $f(\boldsymbol{w}_k)$ of the FOLPP can thus be computed as
$$f(\boldsymbol{w}_k) = \frac{\boldsymbol{w}_k^T\boldsymbol{S}_l\boldsymbol{w}_k}{\boldsymbol{w}_k^T\boldsymbol{S}_d\boldsymbol{w}_k} = \frac{z_k^T\boldsymbol{V}^T\boldsymbol{S}_l\boldsymbol{V}z_k}{z_k^T\boldsymbol{V}^T\boldsymbol{S}_d\boldsymbol{V}z_k}. \tag{17}$$
Noting that $\boldsymbol{S}_l\boldsymbol{V} = \boldsymbol{S}_d\boldsymbol{V}\boldsymbol{\Lambda}$ and $\boldsymbol{V}^T\boldsymbol{S}_d\boldsymbol{V} = \boldsymbol{I}$, we can write Eq. (17) as follows:
$$f(\boldsymbol{w}_k) = \frac{z_k^T\boldsymbol{V}^T\boldsymbol{S}_d\boldsymbol{V}\boldsymbol{\Lambda}z_k}{z_k^T\boldsymbol{V}^T\boldsymbol{S}_d\boldsymbol{V}z_k} = \frac{z_k^T\boldsymbol{\Lambda}z_k}{z_k^Tz_k}. \tag{18}$$
Let $z_k \in \mathbb{R}^{m\times 1} = [z_{1k}, z_{2k}, \ldots, z_{mk}]^T$, such that $f(\boldsymbol{w}_k)$ becomes
$$f(\boldsymbol{w}_k) = \frac{\sum_{i=1}^{m}\lambda_i z_{ik}^2}{\sum_{i=1}^{m}z_{ik}^2}. \tag{19}$$
which reflects the locality preserving power of the FOLPP algorithm. To make sure the FOLPP algorithm has a higher locality preserving power than the LPP algorithm, the value of $f(\boldsymbol{w}_k)$ should satisfy that $f(\boldsymbol{w}_k) \le f(\boldsymbol{v}_k) = \lambda_k$. Let $\boldsymbol{Z}$ be an upper triangular matrix, set $z_k = [z_{1k}, \ldots, z_{kk}, 0, \ldots, 0]^T$

Fig. 2. Sample images of one individual from the Yale data set.

and $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$ in Eq. (19), and we have

$$f(\boldsymbol{w}_k) = \frac{\sum_{i=1}^{k} \lambda_i z_{ik}^2}{\sum_{i=1}^{k} z_{ik}^2} \leq \frac{\lambda_k \sum_{i=1}^{k} z_{ik}^2}{\sum_{i=1}^{k} z_{ik}^2} = \lambda_k. \tag{20}$$

Recalling that $\boldsymbol{Z} = \boldsymbol{Q}\boldsymbol{R}^{-1}$ and $\boldsymbol{R}^{-1}$ is an upper triangular matrix, the locality preserving power of the FOLPP algorithm is determined by the matrix $\boldsymbol{Q}$. To ensure that $\boldsymbol{Z}$ is an upper triangular matrix, note that the product of the two upper triangular matrices is still an upper triangular matrix, and thus that the matrix $\boldsymbol{Q}$ should be chosen as an arbitrary $m \times m$ invertible upper triangular matrix. For simplicity, let $\boldsymbol{Q} = \boldsymbol{I}$ in this paper. Fig. 1 shows the locality preserving power of the LPP, OLPP and FOLPP on the Yale data set (see Section IV-A for details). From Fig. 1, we see that the FOLPP algorithm can have more locality preserving power than the LPP and OLPP algorithms.

IV. Experimental Results

In this section, several experiments are carried out to demonstrate the effectiveness of the proposed FOLPP algorithm. We compare the performance of the proposed algorithm with that of several state-of-the-art DR algorithms, such as PCA [34], LPP [35], and OLPP [28]. For fair comparison, the same graph structures are used for the LPP, the OLPP and the FOLPP algorithms, which are built based on the label information. Moreover, in our comparison we also directly feed the initial high dimensional samples without performing dimensionality reduction to a nearest neighbor (NN) classifier [14], to serve as our baseline method. Experiments have been performed for face recognition on the Yale[1] and AR[2] data sets and for hyperspectral image classification on the Indian Pines[3] and University of Pavia[4] data sets.

A. Experiments on Face Recognition

In the face recognition problem, one is often confronted with the fact that $d \gg n$, where d denotes the dimension of the image vector and n is the number of training image sets. Thus, the $d \times d$ matrix $\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T$ used in LPP, OLPP and FOLPP is singular. Therefore, we can first apply PCA to project the images into a subspace so that the resulting matrix $\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^T$ becomes nonsingular. In addition, PCA is also used as preprocessing for the purposes of noise reduction.

In each experiment, each data set was randomly divided into a training and a testing set with the same numbers. The training set was applied to learn a low dimensional embedding space with different DR algorithms. The testing set was then mapped into the embedding space. After that, we employed an NN classifier based on the Euclidean distance (ℓ_2-norm) to classify the testing set in the embedding space.

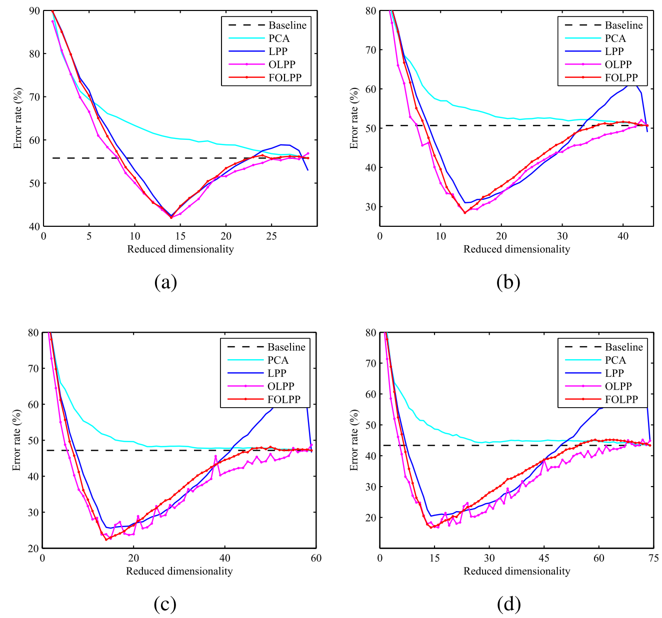


Fig. 3. Error rate versus reduced dimensionality on Yale data set.

1) Face Recognition in the Yale Data Set: The Yale face dataset, which was developed at the Yale Center for Computational Vision and Control, contains 165 gray scale images of 15 individuals. There are 11 images of each individual, which vary in their facial expressions (normal, happy, sad, sleepy, surprised, and winking), illumination conditions and whether the individual appears with/without glasses. Each image is manually cropped and resized to 32×32 pixels [35]. Sample images of one individual from the Yale dataset are displayed in Fig. 2. We randomly split the image samples so that l ($l = 2, 3, 4, 5$) images per individual were taken and labeled to form the training set, while the remainder were used as the testing set. For each given l, we average the results over 30 random splits. In our experiments, the similarity matrix $\boldsymbol{S}$ is defined by the heat kernel function. Empirically, the parameter t is set as the mean norm of the training set.

In general, the performance of all these algorithms varies with the number of dimensions. Table I summarizes the best results and the optimal dimensionality obtained by the Baseline, PCA, LPP, OLPP, and FOLPP algorithms. For the Baseline method, the recognition is simply performed in the original 1024-dimensional image space without any dimensionality reduction. As can be seen from the results,

[1]http://cvc.yale.edu/projects/yalefaces/yalefaces.html
[2]http://rvll.ecn.purdue.edu/~aleix/aleix_face_DB.html
[3]https://engineering.purdue.edu/~biehl/
[4]http://tlclab.unipv.it

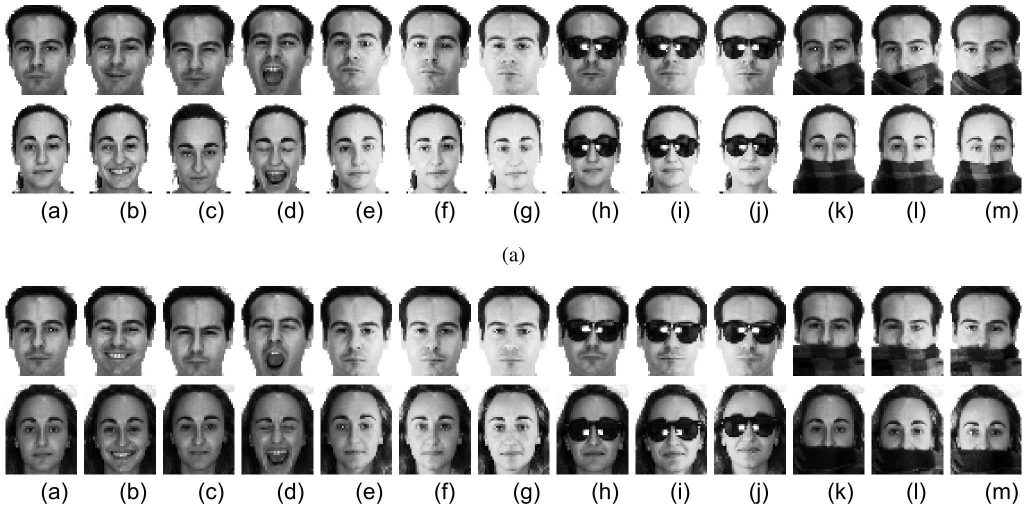


Fig. 4. Sample images of two individuals in the AR data set. (a) Sample images of two individuals in the first session. (b) Sample images of two individuals in the second session.

TABLE I

FACE RECOGNITION ERROR RATES (%) IN THE YALE DATA SET. THE DIMENSION THAT RESULTS IN THE BEST PERFORMANCE FOR EACH ALGORITHM IS SHOWN IN PARENTHESES

Algorithm	2 Train	3 Train	4 Train	5 Train
Baseline	55.78	50.67	47.17	43.33
PCA	55.78 (29)	50.67 (44)	47.08 (58)	43.33 (74)
LPP	42.42 (14)	31.00 (14)	25.59 (15)	20.41 (14)
OLPP	42.00 (14)	28.44 (14)	22.67 (15)	16.78 (16)
FOLPP	**41.98 (14)**	**28.42 (14)**	**22.38 (14)**	**16.70 (14)**

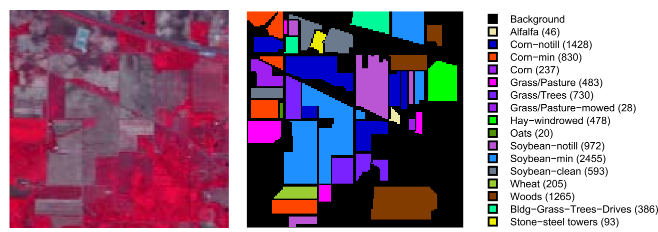
TABLE II

FACE RECOGNITION ERROR RATES (%) IN THE AR DATA SET. THE DIMENSION THAT RESULTS IN THE BEST PERFORMANCE FOR EACH ALGORITHM IS SHOWN IN PARENTHESES

	PCA	LPP	OLPP	FOLPP
Exp 1	14.17 (14)	11.67 (36)	11.67 (31)	**10.83 (31)**
Exp 2	18.21 (26)	16.79 (45)	16.43 (41)	**15.71 (46)**
Exp 3	22.88 (13)	18.27 (43)	18.85 (30)	**18.08 (47)**

our algorithm achieves the best performance. Fig. 3 shows the plots of error rate versus dimensionality reduction. The LPP and OLPP algorithms performed comparatively to our algorithm, while the PCA algorithm performed poorly.

2) Face Recognition in the AR Data Set: The AR face database, established by Purdue University, contains color face images corresponding to 126 different people (70 men and 56 women), depicting their frontal facial view under different facial expressions, illumination conditions and occlusions (sunglasses and scarf). Most subjects were photographed over two sessions (separated by two weeks), so that the database contains two sets of 13 color images (one set for each session) for a total of 120 individuals (65 men and 55 women). Each image is manually cropped and then resized to 50×40. Sample images of two individuals from the two sessions are shown in Fig. 4(a) (first session) and Fig. 4(b) (second session). The images from the first session, namely



(a) (b)

Fig. 5. The Indian Pines hyperspectral image. (a) The HSI in false color. (b) The corresponding ground truth. (Note that the number of each class is shown in brackets.)

TABLE III

TRAINING TIME IN SECONDS REQUIRED BY PCA, LPP, OLPP AND FOLPP ON THE AR DATA SET

Dimension	PCA	LPP	OLPP	FOLPP
100	1.5 s	3.2 s	**120.4** s	**3.7** s
200	1.5 s	3.0 s	**317.9** s	**3.7** s
300	1.8 s	3.5 s	**431.0** s	**3.9** s
400	1.5 s	3.0 s	**917.5** s	**4.3** s
500	2.5 s	4.6 s	**1740.8** s	**5.3** s

(a) "neutral expression," (b) "smile," (c) "anger," (d) "scream," (e) "left light on," (f) "right light on," and (g) "both side lights on" (h) "wearing sunglasses," (i) "wearing sunglasses and left light on," (j) "wearing sunglasses and right light on," (k) "wearing scarf," (l) " wearing scarf and left light on" and (m) "wearing scarf and right light on" are selected as training images, and the corresponding images from the second session are selected for testing.

In order to evaluate the FOLPP algorithm, we have conducted three different experiments with increasing degree of difficulty [36]. For the first experiment (Exp 1), we designed our training set to include only those facial images with illumination variations captured during the first session, while

TABLE IV
CLASSIFICATION RESULTS OF DIFFERENT METHODS AND TRAINING SAMPLES ON THE INDIAN PINES HSI DATA SET

Method	$i = 10\%$			$i = 20\%$		
	AA	OA	κ	AA	OA	κ
Baseline	71.06 (1)	72.95 (1)	0.6909 (1)	75.73 (1)	76.48 (1)	0.7314 (1)
PCA	71.07 (91)	72.93 (97)	0.6906 (97)	75.71 (99)	76.43 (98)	0.7308 (98)
LPP	69.70 (8)	72.51 (9)	0.6868 (9)	76.81 (9)	78.43 (14)	0.7529 (14)
OLPP	58.20 (96)	61.86 (97)	0.5619 (97)	63.36 (80)	65.69 (86)	0.6066 (86)
FOLPP	**80.55 (18)**	**81.10 (18)**	**0.7829 (18)**	**86.61 (17)**	**85.56 (16)**	**0.8348 (16)**
	$i = 30\%$			$i = 40\%$		
	AA	OA	κ	AA	OA	κ
Baseline	78.17 (1)	78.58 (1)	0.7553 (1)	78.85 (1)	79.96 (1)	0.7710 (1)
PCA	78.19 (81)	78.52 (90)	0.7547 (90)	78.82 (100)	79.91 (98)	0.7704 (98)
LPP	81.82 (14)	82.17 (14)	0.7959 (14)	84.15 (14)	83.68 (14)	0.8134 (14)
OLPP	67.34 (77)	68.28 (100)	0.6366 (100)	69.27 (78)	70.29 (79)	0.6598 (79)
FOLPP	**89.17 (15)**	**87.83 (16)**	**0.8609 (16)**	**90.21 (15)**	**89.24 (15)**	**0.877 (15)**

for testing we selected the corresponding images captured during the second recording session. For the second experiment (Exp 2), we used facial images with variation in both illumination conditions and facial expressions from the first session for training and the corresponding images from the second session for testing. Finally, for the third experiment (Exp 3), we used all images from the first session for training and the rest for testing.

Table II summarizes the best results and the optimal dimensionality for each algorithm in each experiment conducted. As can be seen, the FOLPP algorithm performed the best. In Table III we show the recorded CPU training time, which is measured in seconds and required by PCA, LPP, OLPP and FOLPP algorithms in this data set. All algorithms have been implemented on Matlab R2013a and the CPU time required by each algorithm during training has been recorded on a 2.80 GHz computer with 16GB of RAM. It is clear that the computation burden is effectively alleviated for our algorithm compared to the OLPP.

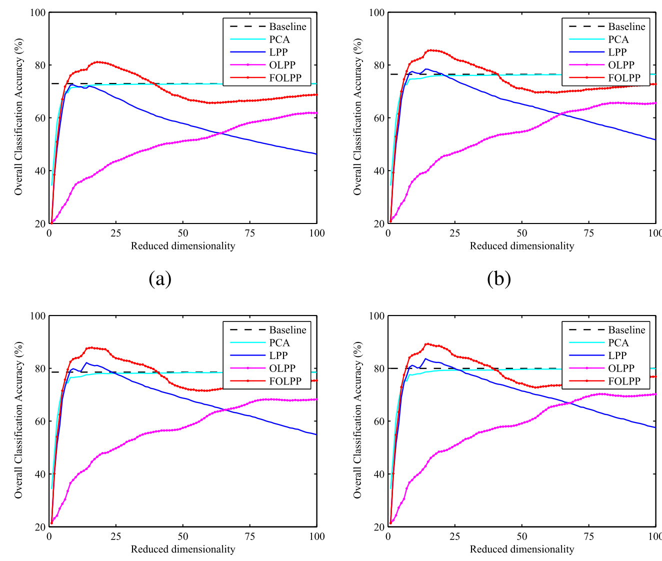
B. Experiments on HSI Classification

In this subsection, classification experiments are conducted on the Indian Pines data set and PaviaU data set to evaluate the performance of the FOLPP algorithm. These two HSI data sets are preprocessed in the following ways: firstly, several spectral bands are removed from the data set due to noise and water absorption, after which the other bands were combined into a high dimensional vector. Each sample was then normalized on a scale of zero to one.

To test these algorithms, each data set was split randomly into a training and a testing set with different numbers. The training set is used to learn a low dimensional embedding space using different DR algorithms, and the testing set is projected onto the embedding space. After that, the NN classifier with Euclidean distance is employed for classification.

To compare the performance of different algorithms, we used the average classification accuracy (AA), the overall classification accuracy (OA), and the kappa coefficient (κ) to assess each algorithm.

1) HSI Classification in the Indian Pines Data Set: The Indian Pines HSI data set [37], [38] was gathered over Northwestern Indiana by the Visible/Infrared Imaging Spectrometer (AVIRIS) sensor in 1992. The AVIRIS sensor generates 220 spectral bands between 0.4 and 2.5 μm. A total of 20 spectral bands with water absorption (104–108, 150–163 and 220) are removed from the data set, leaving 200 bands to be used in the experiments. The image in the data set has spatial resolution of 20m per pixel and spatial dimensions of 145×145 pixels. It contains 16 ground truth classes, most of which are different types of crops (e.g., corns and soybeans). The HSI in false color and its corresponding ground truth are shown in Fig. 5(a) and Fig.5(b), respectively.



(a) (b) (c) (d)

Fig. 6. Overall classification accuracy vs. dimension of different methods on the Indian Pines HSI data set. (a) $i = 10\%$, (b) $i = 20\%$, (c) $i = 30\%$, (d) $i = 40\%$.

In these experiments, we randomly select i ($i = 10\%, 20\%, 30\%, 40\%$) samples of each class for training and the remaining samples are used for testing. For each given i, we average the results over 10 random splits. Fig. 6 shows the OA of various algorithms versus dimensionality reduction under different training conditions. By referring to Fig. 6(a) through Fig. 6(d), we can see that the OA improves with

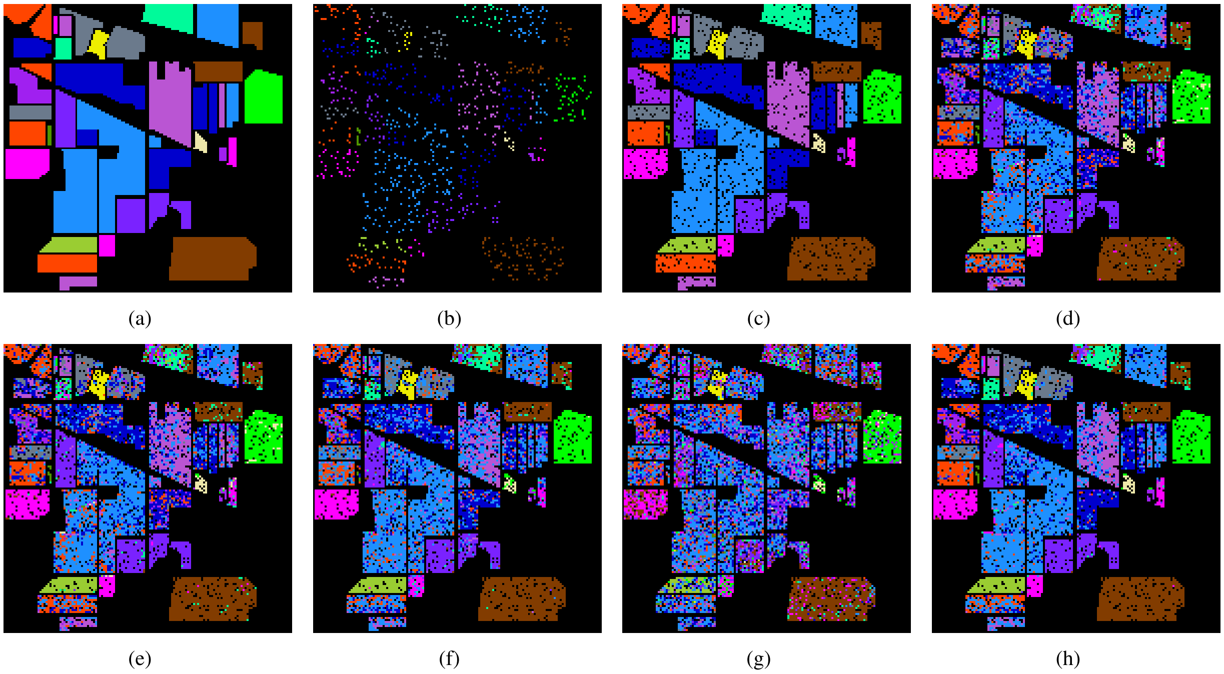


Fig. 7. Classification maps of different methods with NN for Indian Pines HSI. (a) Ground truth, (b) Train samples, (c) Test samples, (d) Baseline (72.63%), (e) PCA (72.05%), (f) LPP (69.94%), (g) OLPP (41.02%), (h) FOLPP (81.93%). Note that OAs are given in parentheses.

TABLE V
CLASSIFICATION RESULTS OF EACH CLASS USING DIFFERENT METHODS ON THE INDIAN PINES HSI DATA SET

Class	Training	Test	Baseline	PCA	LPP	OLPP	FOLPP
Alfalfa	10	36	88.89	88.89	63.89	50.00	**91.67**
Corn-notill	143	1285	59.38	58.52	60.39	27.16	**73.93**
Corn-min	83	747	59.71	60.51	40.43	26.91	**68.01**
Corn	24	213	33.33	32.39	26.29	13.15	**58.69**
Grass/Pasture	48	435	87.13	86.67	84.14	48.28	**92.64**
Grass/Trees	73	657	96.65	96.50	89.19	33.49	**97.56**
Grass/Pasture-mowed	10	18	83.33	83.33	66.67	38.89	**88.89**
Hay-windrowed	48	430	93.95	93.95	99.77	77.44	**100.00**
Oats	10	10	90.00	90.00	40.00	0.00	**100.00**
Soybean-notill	97	875	**65.83**	64.00	42.86	20.23	62.17
Soybean-min	246	2209	70.71	69.62	75.69	50.52	**84.65**
Soybean-clean	59	534	57.49	57.49	54.12	12.17	**76.03**
Wheat	21	184	96.74	96.74	99.46	53.80	**99.46**
Woods	127	1138	94.55	94.38	96.75	73.99	**97.45**
Bldg-Grass-Trees-Drives	39	347	43.52	42.94	53.31	15.27	**67.44**
Stone-steel towers	10	83	**95.18**	**95.18**	91.57	67.47	91.57
AA			76.02	75.69	67.78	38.05	**84.38**
OA			72.63	72.05	69.94	41.02	**81.93**
κ			0.6878	0.6813	0.6513	0.3205	**0.7919**

the increase of the training set and that the proposed FOLPP algorithm outperforms the other algorithms. Table IV summarizes the best results and the optimal dimensionality obtained by all algorithms. As can be seen, for all algorithms, the AAs, OAs and κ coefficients increase as the size of the training set increases. The FOLPP algorithm shows better results compared to the other algorithms.

In order to evaluate the individual classification performance of the FOLPP algorithm, we randomly choose 10% of samples per class for training, while the remaining samples are used for testing. For very small classes, we take a minimum of 10 training samples per class (Alfalfa, Grass/Pasture-mowed, Oats and Stone-steel towers). The embedding dimension was 20-dimensions for all algorithms. Table V shows the classification accuracy of each class, AAs, OAs and κ coefficients for different DR algorithms with the NN classifier. As shown in Table V, the proposed FOLPP algorithm gives the best results when compared to other algorithms.

To visualize the classification results, the classification maps with NN are given in Fig. 7, in which it can be seen that the FOLPP algorithm produces both more homogenous areas and better classification maps than other algorithms, especially in the areas of Grass/Trees, Hay-windrowed, Oats, Wheat and Woods.

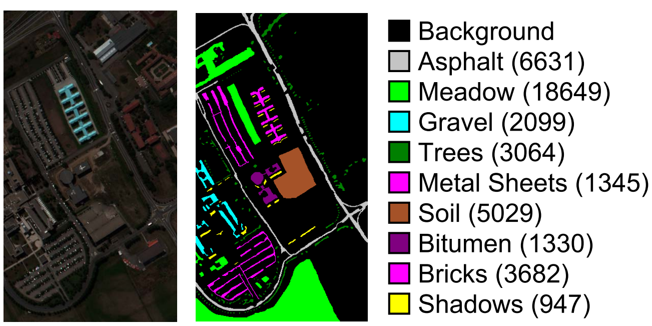
2) HSI Classification in the University of Pavia Data Set: The University of Pavia HSI data set [38], [39] was acquired by the ROSIS sensor during a flight campaign over Pavia University, northern Italy, in 2002. Several spectral bands with

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS AND TRAINING SAMPLES ON THE UNIVERSITY OF PAVIA HSI DATA SET

Method	$i = 5\%$			$i = 10\%$		
	AA	OA	κ	AA	OA	κ
Baseline	79.41 (1)	82.30 (1)	0.7618 (1)	80.80 (1)	83.71 (1)	0.7808 (1)
PCA	79.58 (23)	82.34 (41)	0.7624 (25)	81.00 (27)	83.74 (37)	0.7814 (36)
LPP	79.36 (8)	82.37 (8)	0.7644 (8)	81.36 (8)	83.99 (8)	0.7861 (8)
OLPP	64.49 (60)	70.84 (60)	0.6043 (60)	66.43 (60)	71.81 (60)	0.6181 (60)
FOLPP	**81.16 (10)**	**83.74 (10)**	**0.7819 (10)**	**82.29 (11)**	**84.65 (11)**	**0.7945 (11)**
	$i = 15\%$			$i = 20\%$		
	AA	OA	κ	AA	OA	κ
Baseline	81.71 (1)	84.39 (1)	0.7901 (1)	82.32 (1)	84.93 (1)	0.7974 (1)
PCA	81.90 (27)	84.46 (27)	0.7912 (27)	82.49 (24)	84.98 (25)	0.7984 (25)
LPP	82.24 (8)	84.70 (8)	0.7957 (8)	82.72 (8)	85.03 (9)	0.8001 (9)
OLPP	67.28 (60)	72.48 (60)	0.6274 (60)	66.90 (60)	72.46 (60)	0.6271 (60)
FOLPP	**83.51 (11)**	**85.67 (11)**	**0.8082 (11)**	**83.96 (11)**	**86.05 (11)**	**0.8135 (11)**

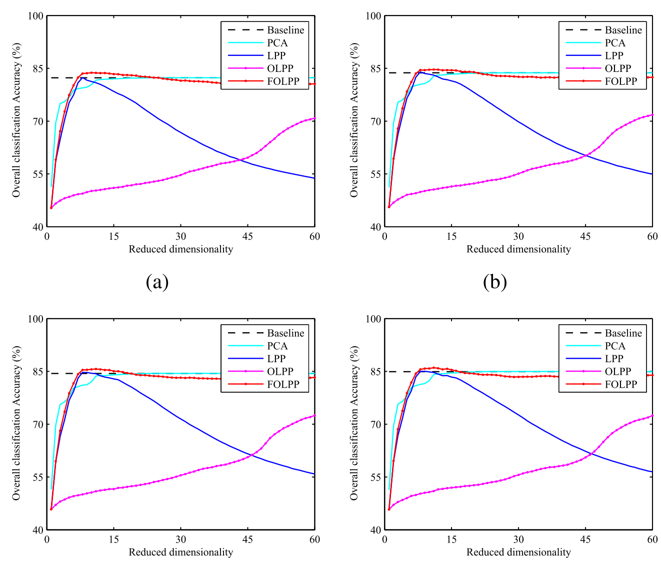
TABLE VII
CLASSIFICATION RESULTS OF EACH CLASS USING DIFFERENT DR METHODS WITH NN ON THE UNIVERSITY OF PAVIA HSI DATA SET

Class	Training	Test	Baseline	PCA	LPP	OLPP	FOLPP
Asphalt	663	5968	80.61	82.71	80.75	35.30	**87.11**
Meadow	1865	16,784	**94.64**	91.53	93.26	73.18	93.61
Gravel	210	1889	64.85	63.95	64.06	16.73	**68.61**
Trees	306	2758	85.71	84.74	**87.82**	41.26	**87.82**
Metal Sheets	135	1210	99.50	99.50	**99.92**	89.59	99.59
Soil	503	4526	56.36	52.10	58.44	16.90	**59.50**
Bitumen	133	1197	76.61	71.01	73.18	20.89	**77.69**
Bricks	368	3314	**74.65**	69.49	69.86	38.56	72.21
Shadows	95	852	97.54	97.89	95.07	59.98	**98.59**
AA			81.16	79.21	80.26	43.60	**82.75**
OA			83.80	81.54	83.01	51.25	**84.91**
κ			0.7822	0.7526	0.7725	0.3472	**0.7980**



(a) (b)

Fig. 8. The University of Pavia hyperspectral image. (a) The HSI in false color. (b) The corresponding ground truth. (Note that the number of each class is shown in brackets.)



(a) (b) (c) (d)

Fig. 9. Overall classification accuracy vs. dimension of different methods on the University of Pavia HSI data set. (a) $i = 5\%$, (b) $i = 10\%$, (c) $i = 15\%$, (d) $i = 20\%$.

noise are removed from the data set, leaving 103 bands to be used in the experiments. The image in the data set has spatial resolution of 1.3 m per pixel and spatial dimensions of 610×340 pixels. It contains 9 ground truth classes: asphalt, meadows, gravel, trees, metal sheets, soil, bitumen, bricks and shadows. For illustrative purposes, the HSI in false color and its corresponding ground truth are shown in Fig. 8(a) and Fig. 8(b), respectively.

In these experiments, we randomly select i ($i = 5\%, 10\%, 15\%, 20\%$) samples of each class for training while the remaining samples are used for testing. For each given i, we average the results over 10 random splits. The OAs versus dimensionality reduction under different training conditions are shown in Fig. 9, and the best results as well as the corresponding dimensions are presented in Table VI. By referring to Fig. 9, we can see that the OA improves with the increase

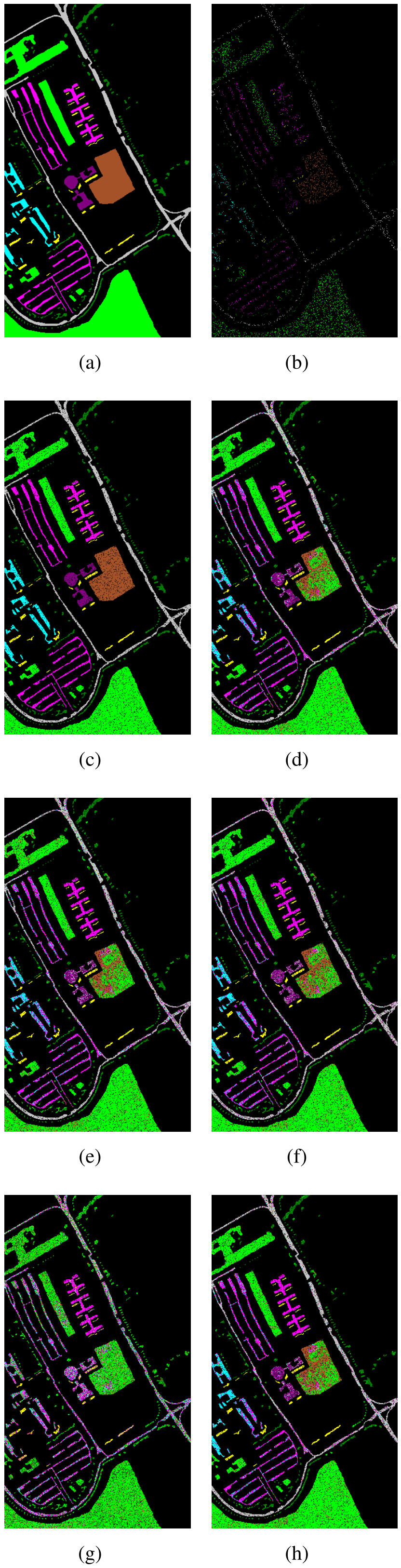


Fig. 10. Classification maps of different methods with NN for University of Pavia HSI. (a) Ground truth, (b) Train samples, (c) Test samples, (d) Baseline (83.80%), (e) PCA (81.54%), (f) LPP (83.01%), (g) OLPP (51.25%), (h) FOLPP (84.91%). Note that OAs are given in parentheses.)

of the training set and the FOLPP algorithm outperforms the other algorithms. As can be seen from Table VI, for all algorithms, the AAs, OAs and κ coefficients increase as the size of the training set increases. It is obvious that the FOLPP algorithm achieves better results compared to other algorithms.

In addition, we randomly select 10% of samples per class for training, while the remaining samples are used for testing. The embedding dimension was 20-dimensions for all algorithms. The NN classifier was used for classification after DR. The classification accuracy of each class, AAs, OAs and κ coefficients are displayed in Table VII and the corresponding classification maps are presented in Fig. 10.

From Table VII, the classification results of FOLPP with NN can be seen to surpass those of Baseline, PCA, LPP and OLPP. In Fig. 10, it can be seen that the FOLPP algorithm produces a smoother classification map than other algorithms in many areas of different classes.

V. Conclusion

Locality preserving projections (LPP) is a linear approximation of Laplacian Eigenmaps with the locality preserving property. However, the projection matrix obtained by LPP is not able to satisfy the desirable orthogonality property. To solve this problem, orthogonal LPP (OLPP) has been proposed; however, this method adopts a step by step procedure, making the algorithm computationally more expensive. Therefore, in this paper, we propose a fast and orthogonal LPP algorithm (FOLPP) to minimize the locality and maximize the globality simultaneously under an orthogonal projection matrix. As a result, the computation burden of the proposed algorithm is effectively reduced compared to the OLPP algorithm. Experimental results on face recognition and HSI classification demonstrate the effectiveness and superiority of the proposed algorithm.

References

[1] M. Xu, H. Chen, and P. Varshney, "Dimensionality reduction for registration of high-dimensional data sets," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3041–3049, Aug. 2013.

[2] T. Zhou and D. Tao, "Double shrinking sparse dimension reduction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 244–257, Jan. 2013.

[3] X. Liu, L. Huang, C. Deng, B. Lang, and D. Tao, "Query-adaptive hash code ranking for large-scale multi-view visual search," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4514–4524, Oct. 2016.

[4] Y. Yang, C. Deng, S. Gao, W. Liu, D. Tao, and X. Gao, "Discriminative multi-instance multitask learning for 3D action recognition," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 519–529, Mar. 2017.

[5] Y. Yang, C. Deng, D. Tao, S. Zhang, W. Liu, and X. Gao, "Latent max-margin multitask learning with skelets for 3-D action recognition," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 439–448, Feb. 2017.

[6] Y. Huang, D. Xu, and F. Nie, "Semi-supervised dimension reduction using trace ratio criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 519–526, Mar. 2012.

[7] Q. Gao, J. Ma, H. Zhang, X. Gao, and Y. Liu, "Stable orthogonal local discriminant embedding for linear dimensionality reduction," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2521–2531, Jul. 2013.

[8] S.-J. Wang, S. Yan, J. Yang, C.-G. Zhou, and X. Fu, "A general exponential framework for dimensionality reduction," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 920–930, Feb. 2014.

[9] R. Wang, F. Nie, X. Yang, F. Gao, and M. Yao, "Robust 2DPCA with non-greedy ℓ_1-norm maximization for image analysis," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1108–1112, May 2015.

[10] Q. Yu, R. Wang, X. Yang, B. N. Li, and M. Yao, "Diagonal principal component analysis with non-greedy ℓ_1-norm maximization for face recognition," *Neurocomputing*, vol. 171, pp. 57–62, Jan. 2016.

[11] X. Chang, F. Nie, Y. Yang, C. Zhang, and H. Huang, "Convex sparse PCA for unsupervised feature learning," *ACM Trans. Knowl. Discovery Data*, vol. 11, no. 1, pp. 3:1–3:16, Jul. 2016.

[12] X. Chang, F. Nie, S. Wang, Y. Yang, X. Zhou, and C. Zhang, "Compound rank-k projections for bilinear analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 7, pp. 1502–1513, Jul. 2016.

[13] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2002.

[14] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[15] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[16] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[17] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, vol. 14. Cambridge, MA, USA: MIT Press, 2001, pp. 585–591.

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

[19] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 830–837.

[20] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[21] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao, "Visual reranking through weakly supervised multi-graph learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2600–2607.

[22] C. Deng, R. Ji, D. Tao, X. Gao, and X. Li, "Weakly supervised multi-graph learning for robust image reranking," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 785–795, Apr. 2014.

[23] C. Deng, X. Tang, J. Yan, W. Liu, and X. Gao, "Discriminative dictionary learning with common label alignment for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 208–218, Feb. 2016.

[24] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems*, vol. 16. Cambridge, MA, USA: MIT Press, 2003, pp. 153–160.

[25] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 1208–1213.

[26] D. Cai, X. He, and J. Han, "Isometric projection," in *Proc. Assoc. Adv. Artif. Intell.*, 2007, pp. 528–533.

[27] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, Dec. 2007.

[28] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.

[29] J. Duchene and S. Leclercq, "An optimal transformation for discriminant and principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 978–983, Nov. 1988.

[30] F. Nie, S. Xiang, Y. Song, and C. Zhang, "Orthogonal locality minimizing globality maximizing projections for feature extraction," *Opt. Eng.*, vol. 48, no. 1, pp. 017202-1–017202-5, 2009.

[31] Y. Jia, F. Nie, and C. Zhang, "Trace ratio problem revisited," *IEEE Trans. Neural Netw.*, vol. 20, no. 4, pp. 729–735, Apr. 2009.

[32] F. Nie, S. Xiang, Y. Jia, and C. Zhang, "Semi-supervised orthogonal discriminant analysis via label propagation," *Pattern Recognit.*, vol. 42, no. 11, pp. 2615–2627, 2009.

[33] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample and out-of-sample spectral clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1796–1808, Nov. 2011.

[34] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1991, pp. 586–591.

[35] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[36] S. Nikitidis, A. Tefas, and I. Pitas, "Maximum margin projection subspace learning for visual data analysis," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4413–4425, Oct. 2014.

[37] E. Zhang, X. Zhang, S. Yang, and S. Wang, "Improving hyperspectral image classification using spectral information divergence," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 249–253, Jan. 2014.

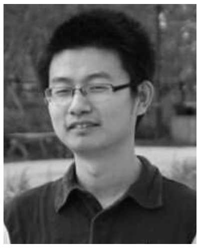
[38] H. Huang, F. Luo, J. Liu, and Y. Yang, "Dimensionality reduction of hyperspectral images based on sparse discriminant manifold embedding," *J. Photogramm. Remote Sens.*, vol. 106, pp. 42–54, Aug. 2015.

[39] H. Yuan, Y. Y. Tang, Y. Lu, L. Yang, and H. Luo, "Hyperspectral image classification based on regularized sparse representation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2174–2182, Jun. 2014.

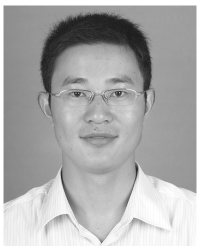
Rong Wang (M'12) received the B.E. degree in information engineering, the M.E. degree in signal and information processing, and the Ph.D. degree in computer science from the Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007, and 2013, respectively. From 2007 to 2013, he was also with the Department of Automation, Tsinghua University, Beijing, China, for the Ph.D. degree. His current research interests include machine learning and signal processing, together with their applications including pattern recognition, image processing, and computer vision.

Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, China, in 2009. He is currently a Full Professor with Northwestern Polytechnical University, China. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He has authored over 100 papers in the following top journals and conferences: TPAMI, IJCV, TIP, TNNLS/TNN, TKDE, Bioinformatics, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited over 7000 times. He is serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields.

Richang Hong (M'12) received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008. He was a Research Fellow with the School of Computing, National University of Singapore, from 2008 to 2010. He is currently a Professor with the Hefei University of Technology, Hefei, China. He has co-authored over 70 publications in his research interests, which include multimedia content analysis and social media. He is a member of the ACM and the Executive Committee Member of the ACM SIGMM China Chapter. He was a recipient of the Best Paper Award in the ACM Multimedia 2010, the Best Paper Award in the ACM ICMR 2015 and the Honorable Mention of the IEEE TRANSACTIONS ON MULTIMEDIA Best Paper Award. He served as an Associate Editor of the *Information Sciences* and *Signal Processing* Elsevier, and the Technical Program Chair of the MMM 2016.



Xiaojun Chang received the Ph.D. degree in computer science from the Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Australia, in 2016. He has been a Post-Doctoral Research Associate with the Language Technology Institute, Carnegie Mellon University. His main research interests include machine learning, data mining, and computer vision.



Xiaojun Yang received the Ph.D. degree from the High-Tech Institute of Xi'an, Xi'an, China, in 2010. He is currently an Associate Professor with the Guangdong University of Technology. His research interests include machine learning, signal processing, and data fusion.

Weizhong Yu received the Ph.D. degree from the High-Tech Institute of Xi'an, China, in 2012. He is currently a Post-Doctoral Research Associate with the School of Electronics and Information Engineering, Xi'an Jiaotong University. His main research interests include machine learning and computer vision.