[29] G. Deng and J. C. Pinoli, "Differentiation-based edge detection using the logarithmic image processing model," *J. Math. Imag. Vis.*, vol. 8, no. 2, pp. 161–180, Mar. 1998.
[30] V. Patrascu and V. Buzuloiu, "Image dynamic range enhancement in the context of logarithmic models," in *Proc. 11th Eur. Signal Process.*, Toulouse, France, Sep. 2002, vol. 3, pp. 251–254.
[31] W. Rudin, *Real and Complex Analysis (Higher Mathematics Series)*, 3rd ed. New York: McGraw-Hill, 1986.

Orthogonal Laplacianfaces for Face Recognition

Deng Cai, Xiaofei He, Jiawei Han, *Senior Member, IEEE*, and Hong-Jiang Zhang, *Fellow, IEEE*

***Abstract*—Following the intuition that the naturally occurring face data may be generated by sampling a probability distribution that has support on or near a submanifold of ambient space, we propose an appearance-based face recognition method, called orthogonal Laplacianface. Our algorithm is based on the locality preserving projection (LPP) algorithm, which aims at finding a linear approximation to the eigenfunctions of the Laplace Beltrami operator on the face manifold. However, LPP is nonorthogonal, and this makes it difficult to reconstruct the data. The orthogonal locality preserving projection (OLPP) method produces orthogonal basis functions and can have more locality preserving power than LPP. Since the locality preserving power is potentially related to the discriminating power, the OLPP is expected to have more discriminating power than LPP. Experimental results on three face databases demonstrate the effectiveness of our proposed algorithm.**

***Index Terms*—Appearance-based vision, face recognition, locality preserving projection (LPP), orthogonal locality preserving projection (OLPP).**

I. Introduction

Recently, appearance-based face recognition has received a lot of attention [20], [14]. In general, a face image of size $n_1 \times n_2$ is represented as a vector in the image space $\mathbb{R}^{n_1 \times n_2}$. We denote by face space the set of all the face images. Though the image space is very high dimensional, the face space is usually a submanifold of very low dimensionality which is embedded in the ambient space. A common way to attempt to resolve this problem is to use dimensionality reduction techniques [1], [2], [8], [11], [12], [17]. The most popular methods discovering the face manifold structure include Eigenface [20], Fisherface [2], and Laplacianface [9].

Face representation is fundamentally related to the problem of manifold learning [3], [16], [19] which is an emerging research area. Given a set of high-dimensional data points, manifold learning techniques aim at discovering the geometric properties of the data space, such as its Euclidean embedding, intrinsic dimensionality, connected components, homology, etc. Particularly, learning representation is closely related to the embedding problem, while clustering can be thought of as finding connected components. Finding a Euclidean embedding of the face space for recognition is the primary focus of our work in this paper. Manifold learning techniques can be classified into linear and nonlinear techniques. For face processing, we are especially interested in linear techniques due to the consideration of computational complexity.

The Eigenface and Fisherface methods are two of the most popular linear techniques for face recognition. Eigenface applies Principal Component Analysis (PCA) [6] to project the data points along the directions of maximal variances. The Eigenface method is guaranteed to discover the intrinsic geometry of the face manifold when it is linear. Unlike the Eigenface method which is unsupervised, the Fisherface method is supervised. Fisherface applies Linear Discriminant Analysis (LDA) to project the data points along the directions optimal for discrimination. Both Eigenface and Fisherface see only the global Euclidean structure. The Laplacianface method [9] is recently proposed to model the local manifold structure. The Laplacianfaces are the linear approximations to the eigenfunctions of the Laplace Beltrami operator on the face manifold. However, the basis functions obtained by the Laplacianface method are nonorthogonal. This makes it difficult to reconstruct the data.

In this paper, we propose a new algorithm called **orthogonal Laplacianface**. O-Laplacianface is fundamentally based on the Laplacianface method. It builds an adjacency graph which can best reflect the geometry of the face manifold and the class relationship between the sample points. The projections are then obtained by preserving such a graph structure. It shares the same locality preserving character as Laplacianface, but at the same time it requires the basis functions to be orthogonal. Orthogonal basis functions preserve the metric structure of the face space. In fact, if we use all the dimensions obtained by O-Laplacianface, the projective map is simply a rotation map which does not distort the metric structure. Moreover, our empirical study shows that O-Laplacianface can have more locality preserving power than Laplacianface. Since it has been shown that the locality preserving power is directly related to the discriminating power [9], the O-Laplacianface is expected to have more discriminating power than Laplacianface.

The rest of the paper is organized as follows. In Section II, we give a brief review of the Laplacianface algorithm. Section III introduces our O-Laplacianface algorithm. We provide a theoretical justification of our algorithm in Section IV. Extensive experimental results on face recognition are presented in Section V. Finally, we provide some concluding remarks and suggestions for future work in Section VI.

II. Brief Review of Laplacianface

Laplacianface is a recently proposed linear method for face representation and recognition. It is based on locality preserving projection [10] and explicitly considers the manifold structure of the face space.

Given a set of face images $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^m$, let $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$. Let S be a similarity matrix defined on the data points. Laplacianface can be obtained by solving the following minimization problem:

$$\begin{aligned}\mathbf{a}_{opt} &= \arg\min_{\mathbf{a}} \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\mathbf{a}^T\mathbf{x}_i - \mathbf{a}^T\mathbf{x}_j\right)^2 S_{ij} \\ &= \arg\min_{\mathbf{a}} \mathbf{a}^T X L X^T \mathbf{a}\end{aligned}$$

Manuscript received August 30, 2005; revised March 22, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Manuel Sameulides.

D. Cai and J. Han are with the Department of Computer Science, University of Illinois at Urbana Champaign, Urbana, IL 61801 USA (e-mail: dengcai2@cs.uiuc.edu; hanj@cs.uiuc.edu).

X. He is with Yahoo Research Labs, Burbank, CA 91504 USA (e-mail: hex@yahoo-inc.com).

H.-J. Zhang is with Microsoft Research Asia, Beijing 100080, China (e-mail: hjzhang@microsoft.com).

Color versions of Figs. 1 and 3–5 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.881945

with the constraint

$$\mathbf{a}^T X D X^T \mathbf{a} = 1$$

where $L = D - S$ is the *graph Laplacian* [4] and $D_{ii} = \sum_j S_{ij}$. D_{ii} measures the local density around $\mathbf{x}_i$. Laplacianface constructs the similarity matrix S as shown in the equation at the bottom of the next page. Here, S_{ij} is actually heat kernel weight, the justification for such choice and the setting of the parameter t can be referred to [3].

The objective function in Laplacianface incurs a heavy penalty if neighboring points $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped far apart. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close," then $y_i \left(= \mathbf{a}^T \mathbf{x}_i\right)$ and $y_j \left(= \mathbf{a}^T \mathbf{x}_j\right)$ are close, as well [9]. Finally, the basis functions of Laplacianface are the eigenvectors associated with the smallest eigenvalues of the following generalized eigen-problem:

$$X L X^T \mathbf{a} = \lambda X D X^T \mathbf{a}$$

XDX^T is nonsingular after some preprocessing steps on X in Laplacianface; thus, the basis functions of Laplacianface can also be regarded as the eigenvectors of the matrix $(XDX^T)^{-1} XLX^T$ associated with the smallest eigenvalues. Since $(XDX^T)^{-1} XLX^T$ is not symmetric in general, the basis functions of Laplacianface are nonorthogonal.

Once the eigenvectors are computed, let $A_k = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$ be the transformation matrix. Thus, the Euclidean distance between two data points in the reduced space can be computed as follows:

$$\begin{aligned}
\operatorname{dist}(\mathbf{y}_i, \mathbf{y}_j) &= \|\mathbf{y}_i - \mathbf{y}_j\| \\
&= \left\| A^T \mathbf{x}_i - A^T \mathbf{x}_j \right\| \\
&= \left\| A^T (\mathbf{x}_i - \mathbf{x}_j) \right\| \\
&= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T A A^T (\mathbf{x}_i - \mathbf{x}_j)}.
\end{aligned}$$

If A is an orthogonal matrix, $AA^T = I$ and the metric structure is preserved.

III. Algorithm

In this section, we introduce a novel subspace learning algorithm, called Orthogonal Locality Preserving Projection (OLPP). Our orthogonal Laplacianface algorithm for face representation and recognition is based on OLPP. The theoretical justifications of our algorithm will be presented in Section IV.

In appearance-based face analysis one is often confronted with the fact that the dimension of the face image vector (m) is much larger than the number of face images (n). Thus, the $m \times m$ matrix XDX^T is singular. To overcome this problem, we can first apply PCA to project the faces into a subspace without losing any information and the matrix XDX^T becomes nonsingular.

The algorithmic procedure of OLPP is stated as follows.

1) **PCA Projection**: We project the face images x_i into the PCA subspace by throwing away the components corresponding to zero eigenvalue. We denote the transformation matrix of PCA by W_{PCA}. By PCA projection, the extracted features are statistically uncorrelated and the rank of the new data matrix is equal to the number of features (dimensions).

2) **Constructing the Adjacency Graph**: Let G denote a graph with n nodes. The ith node corresponds to the face image $\mathbf{x}_i$. We put an edge between nodes i and j if $\mathbf{x}_i$ and $\mathbf{x}_j$ are "close," i.e., $\mathbf{x}_i$ is among p nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among p nearest neighbors of $\mathbf{x}_i$. Note that, if the class information 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, put

$$S_{ij} = e^{-\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t\right)}.$$

Otherwise, put $S_{ij} = 0$. The weight matrix S of graph G models the local structure of the face manifold. The justification of this weight can be traced back to **[3]**.

4) **Computing the Orthogonal Basis Functions**: We define D as a diagonal matrix whose entries are column (or row, since S is symmetric) sums of S, $D_{ii} = \sum_j S_{ji}$. We also define $L = D - S$, which is called Laplacian matrix in spectral graph theory **[4]**. Let $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k\}$ be the orthogonal basis vectors, we define

$$\begin{aligned}
A^{(k-1)} &= [\mathbf{a}_1, \ldots, \mathbf{a}_{k-1}] \\
B^{(k-1)} &= \left[A^{(k-1)}\right]^T (XDX^T)^{-1} A^{(k-1)}.
\end{aligned}$$

The orthogonal basis vectors $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_k\}$ can be computed as follows.

- Compute $\mathbf{a}_1$ as the eigenvector of $(XDX^T)^{-1} XLX^T$ associated with the smallest eigenvalue.
- Compute $\mathbf{a}_k$ as the eigenvector of

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

associated with the smallest eigenvalue of $M^{(k)}$.

5) **OLPP Embedding**: Let $W_{\mathrm{OLPP}} = [\mathbf{a}_1, \ldots, \mathbf{a}_l]$, the embedding is as follows:

$$\begin{aligned}
\mathbf{x} \rightarrow \mathbf{y} &= W^T \mathbf{x} \\
W &= W_{\mathrm{PCA}} W_{\mathrm{OLPP}}
\end{aligned}$$

where $\mathbf{y}$ is a l-dimensional representation of the face image $\mathbf{x}$, and W is the transformation matrix.

IV. Justifications

In this section, we provide theoretical justifications of our proposed algorithm.

$$S_{ij} = \begin{cases} e^{-\left(\|\mathbf{x}_i - \mathbf{x}_j\|^2 / t\right)}, & \text{if } \mathbf{x}_i \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_j \text{ or } \mathbf{x}_j \text{ is among the } p \text{ nearest neighbors of } \mathbf{x}_i \\ 0, & \text{otherwise} \end{cases}$$

A. Optimal Orthogonal Embedding

We begin with the following definition.

Definition: Let $\mathbf{a} \in \mathbb{R}^m$ be a projective map. The **locality preserving function** f is defined as follows:

$$f(\mathbf{a}) = \frac{\mathbf{a}^T XLX^T \mathbf{a}}{\mathbf{a}^T XDX^T \mathbf{a}}. \quad (1)$$

Consider the data are sampled from an underlying data manifold $\mathcal{M}$. Suppose we have a map $g : \mathcal{M} \to \mathbb{R}$. The gradient $\nabla g(\mathbf{x})$ is a vector field on the manifold, such that for small $\delta\mathbf{x}$

$$|g(\mathbf{x} + \delta\mathbf{x}) - g(\mathbf{x})| \approx |\langle \nabla g(\mathbf{x}), \delta\mathbf{x} \rangle| \leq \|\nabla g\| \|\delta\mathbf{x}\|.$$

Thus, we see that if $\|\nabla g\|$ is small, points near $\mathbf{x}$ will be mapped to points near $g(\mathbf{x})$. We can use

$$\frac{\int_{\mathcal{M}} \|\nabla g(\mathbf{x})\|^2 d\mathbf{x}}{\int_{\mathcal{M}} |g(\mathbf{x})|^2 d\mathbf{x}} \quad (2)$$

to measure the locality preserving power on average of the map g [3]. With finite number of samples X and a linear projective map $\mathbf{a}$, $f(\mathbf{a})$ is a discrete approximation of equation (2) [10]. Similarly, $f(\mathbf{a})$ evaluates the locality preserving power of the projective map $\mathbf{a}$.

Directly minimizing the function $f(\mathbf{a})$ will lead to the original Laplacianface (LPP) algorithm. Our O-Laplacianface (OLPP) algorithm tries to find a set of orthogonal basis vectors $\mathbf{a}_1, \ldots, \mathbf{a}_k$ which minimizes the locality preserving function. Thus, $\mathbf{a}_1, \ldots, \mathbf{a}_k$ are the set of vectors minimizing $f(\mathbf{a})$ subject to the constraint $\mathbf{a}_k^T \mathbf{a}_1 = \mathbf{a}_k^T \mathbf{a}_2 = \cdots = \mathbf{a}_k^T \mathbf{a}_{k-1} = 0$.

The objective function of OLPP can be written as

$$\mathbf{a}_1 = \arg\min_{\mathbf{a}} \frac{\mathbf{a}^T XLX^T \mathbf{a}}{\mathbf{a}^T XDX^T \mathbf{a}} \quad (3)$$

and

$$\mathbf{a}_k = \arg\min_{\mathbf{a}} \frac{\mathbf{a}^T XLX^T \mathbf{a}}{\mathbf{a}^T XDX^T \mathbf{a}}$$
$$\text{subject to } \mathbf{a}_k^T \mathbf{a}_1 = \mathbf{a}_k^T \mathbf{a}_2 = \cdots = \mathbf{a}_k^T \mathbf{a}_{k-1} = 0. \quad (4)$$

Since XDX^T is positive definite after PCA projection, for any $\mathbf{a}$, we can always normalize it such that $\mathbf{a}^T XDX^T \mathbf{a} = 1$, and the ratio of $\mathbf{a}^T XLX^T \mathbf{a}$ and $\mathbf{a}^T XDX^T \mathbf{a}$ remains unchanged. Thus, the above minimization problem is equivalent to minimizing the value of $\mathbf{a}^T XLX^T \mathbf{a}$ with an additional constraint as follows:

$$\mathbf{a}^T XDX^T \mathbf{a} = 1.$$

Note that, the above normalization is only for simplifying the computation. Once we get the optimal solutions, we can re-normalize them to get an orthonormal basis vectors.

It is easy to check that $\mathbf{a}_1$ is the eigenvector of the generalized eigenproblem

$$XLX^T \mathbf{a} = \lambda XDX^T \mathbf{a}$$

associated with the smallest eigenvalue. Since XDX^T is nonsingular, $\mathbf{a}_1$ is the eigenvector of the matrix $(XDX^T)^{-1} XLX^T$ associated with the smallest eigenvalue.

In order to get the kth basis vector, we minimize the following objective function

$$f(\mathbf{a}_k) = \frac{\mathbf{a}_k^T XLX^T \mathbf{a}_k}{\mathbf{a}_k^T XDX^T \mathbf{a}_k} \quad (5)$$

with the constraints

$$\mathbf{a}_k^T \mathbf{a}_1 = \mathbf{a}_k^T \mathbf{a}_2 = \cdots = \mathbf{a}_k^T \mathbf{a}_{k-1} = 0, \quad \mathbf{a}_k^T XDX^T \mathbf{a}_k = 1.$$

We can use the Lagrange multipliers to transform the above objective function to include all the constraints

$$\begin{aligned} C^{(k)} &= \mathbf{a}_k^T XLX^T \mathbf{a}_k \\ &\quad -\lambda \left(\mathbf{a}_k^T XDX^T \mathbf{a}_k - 1 \right) - \mu_1 \mathbf{a}_k^T \mathbf{a}_1 - \cdots - \mu_{k-1} \mathbf{a}_k^T \mathbf{a}_{k-1}. \end{aligned}$$

The optimization is performed by setting the partial derivative of $C^{(k)}$ with respect to $\mathbf{a}_k$ to zero

$$\begin{aligned} &\frac{\partial C^{(k)}}{\partial \mathbf{a}_k} = 0 \\ &\qquad \Rightarrow 2XLX^T \mathbf{a}_k - 2\lambda XDX^T \mathbf{a}_k \\ &\qquad\quad - \mu_1 \mathbf{a}_1 \ldots - \mu_{k-1} \mathbf{a}_{k-1} = 0. \end{aligned} \quad (6)$$

Multiplying the left side of (6) by $\mathbf{a}_k^T$, we obtain

$$\begin{aligned} &2\mathbf{a}_k^T XLX^T \mathbf{a}_k - 2\lambda \mathbf{a}_k^T XDX^T \mathbf{a}_k = 0 \\ &\qquad \Rightarrow \lambda = \frac{\mathbf{a}_k^T XLX^T \mathbf{a}_k}{\mathbf{a}_k^T XDX^T \mathbf{a}_k}. \end{aligned} \quad (7)$$

Comparing to (5), λ exactly represents the expression to be minimized.

Multiplying the left side of (6) successively by $\mathbf{a}_1^T (XDX^T)^{-1}, \ldots, \mathbf{a}_{k-1}^T (XDX^T)^{-1}$, we now obtain a set of $k-1$ equations

$$\begin{aligned} &\mu_1 \mathbf{a}_1^T (XDX^T)^{-1} \mathbf{a}_1 + \cdots + \mu_{k-1} \mathbf{a}_1^T (XDX^T)^{-1} \mathbf{a}_{k-1} \\ &\quad = 2\mathbf{a}_1^T (XDX^T)^{-1} XLX^T \mathbf{a}_k \\ &\mu_1 \mathbf{a}_2^T (XDX^T)^{-1} \mathbf{a}_1 + \cdots + \mu_{k-1} \mathbf{a}_2^T (XDX^T)^{-1} \mathbf{a}_{k-1} \\ &\quad = 2\mathbf{a}_2^T (XDX^T)^{-1} XLX^T \mathbf{a}_k \\ &\quad \cdots\cdots \\ &\mu_1 \mathbf{a}_{k-1}^T (XDX^T)^{-1} \mathbf{a}_1 + \cdots + \mu_{k-1} \mathbf{a}_{k-1}^T (XDX^T)^{-1} \mathbf{a}_{k-1} \\ &\quad = 2\mathbf{a}_{k-1}^T (XDX^T)^{-1} XLX^T \mathbf{a}_k. \end{aligned}$$

We define

$$\begin{aligned} &\mu^{(k-1)} = [\mu_1, \ldots, \mu_{k-1}]^T, \quad A^{(k-1)} = [\mathbf{a}_1, \ldots, \mathbf{a}_{k-1}] \\ &B^{(k-1)} = \left[B_{ij}^{(k-1)} \right] = \left[A^{(k-1)} \right]^T (XDX^T)^{-1} A^{(k-1)} \\ &B_{ij}^{(k-1)} = \mathbf{a}_i^T (XDX^T)^{-1} \mathbf{a}_j. \end{aligned}$$

Using this simplified notation, the previous set of $k-1$ equations can be represented in a single matrix relationship

$$B^{(k-1)} \mu^{(k-1)} = 2 \left[A^{(k-1)} \right]^T (XDX^T)^{-1} XLX^T \mathbf{a}_k$$

thus

$$\mu^{(k-1)} = 2 \left[B^{(k-1)} \right]^{-1} \left[A^{(k-1)} \right]^T (XDX^T)^{-1} XLX^T \mathbf{a}_k. \quad (8)$$

Let us now multiply the left side of (6) by $(XDX^T)^{-1}$

$$\begin{aligned} &2(XDX^T)^{-1} XLX^T \mathbf{a}_k - 2\lambda \mathbf{a}_k \\ &\qquad -\mu_1 (XDX^T)^{-1} \mathbf{a}_1 - \cdots - \mu_{k-1} (XDX^T)^{-1} \mathbf{a}_{k-1} = 0. \end{aligned}$$

This can be expressed using matrix notation as

$$2(XDX^T)^{-1}XLX^T\mathbf{a}_k - 2\lambda\mathbf{a}_k - (XDX^T)^{-1}A^{(k-1)}\mu^{(k-1)} = 0.$$

With (8), we obtain

$$\left\{I - (XDX^T)^{-1}A^{(k-1)}\left[B^{(k-1)}\right]^{-1}\left[A^{(k-1)}\right]^T\right\} \cdot (XDX^T)^{-1}XLX^T\mathbf{a}_k = \lambda\mathbf{a}_k.$$

As shown in (7), λ is just the criterion to be minimized, thus $\mathbf{a}_k$ is the eigenvector of

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

associated with the smallest eigenvalue of $M^{(k)}$.

Finally, we get the optimal orthogonal basis vectors. The orthogonal basis of O-Laplacianface preserves the metric structure of the face space. It would be important to note that the derivation presented here is motivated by [5].

Recall in the Laplacianface method [9], the basis vectors are the first k eigenvectors associated with the smallest eigenvalues of the eigen-problem

$$XLX^T\mathbf{b} = \lambda XDX^T\mathbf{b}. \tag{9}$$

Thus, the basis vectors satisfy the following equation:

$$\mathbf{b}_i^T XDX^T\mathbf{b}_j = 0 \quad (i \neq j).$$

Clearly, the transformation of the Laplacianface (LPP) method is nonorthogonal. In fact, it is XDX^T-orthogonal.

B. Locality Preserving Power

Both LPP and OLPP try to preserve the local geometric structure. They find the basis vectors by minimizing the locality preserving function

$$f(\mathbf{a}) = \frac{\mathbf{a}^T XLX^T\mathbf{a}}{\mathbf{a}^T XDX^T\mathbf{a}} \tag{10}$$

$f(\mathbf{a})$ reflects the locality preserving power of the projective map $\mathbf{a}$.

In the LPP algorithm, based on the Rayleigh quotient format of the eigen-problem (9) [7], the value of $f(\mathbf{a})$ is exactly the eigenvalue of (9) corresponding to eigenvector $\mathbf{a}$. Therefore, the eigenvalues of LPP reflect the locality preserving power of LPP. In OLPP, as we show in (7), the eigenvalues of OLPP also reflect its locality preserving power. This observation motivates us to compare the eigenvalues of LPP and OLPP.

Fig. 1 shows the eigenvalues of LPP and OLPP. The data set used for this study is the ORL face database (please see Section V-B for details). As can be seen, the eigenvalues of OLPP are consistently smaller than those of LPP, which indicates that OLPP can have more locality preserving power than LPP.

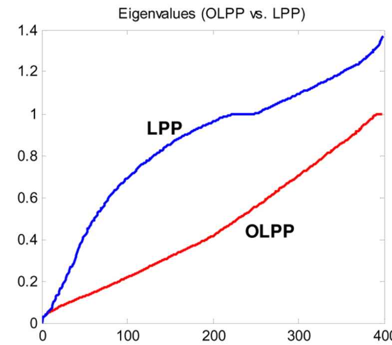

Fig. 1. Eigenvalues of LPP and OLPP. As can be seen, the eigenvalues of OLPP are consistently smaller than those of LPP, which indicates that OLPP can have more locality preserving power than LPP.

Since it has been shown in [9] that the locality preserving power is directly related to the discriminating power, we expect that the O-Laplacianface (OLPP) based face representation and recognition can obtain better performance than those based on Laplacianface (LPP).

V. Experimental Results

In this section, we investigate the performance of our proposed O-Laplacianface method $(\mathrm{PCA}+\mathrm{OLPP})$ for face representation and recognition. The system performance is compared with the Eigenface method (PCA) [21], the Fisherface method $(\mathrm{PCA}+\mathrm{LDA})$ [2], and the Laplacianface method $(\mathrm{PCA} + \mathrm{LPP})$ [9], three of the most popular linear methods in face recognition. We use the same graph structures in the Laplacianface and O-Laplacianface methods, which is built based on the label information.

In this study, three face databases were tested. The first one is the Yale database,[1] the second is the Olivetti Research Laboratory (ORL) database,[2] and the third is the PIE (pose, illumination, and expression) database from CMU [18]. In all the experiments, preprocessing to locate the faces was applied. Original images were manually aligned (two eyes were aligned at the same position), cropped, and then re-sized to 32×32 pixels, with 256 gray levels per pixel. Each image is represented by a 1,024-dimensional vector in image space. Different pattern classifiers have been applied for face recognition, such as nearest-neighbor [2], Bayesian [13], and support vector machine [15]. In this paper, we apply the nearest-neighbor classifier for its simplicity. The Euclidean metric is used as our distance measure.

In short, the recognition process has three steps. First, we calculate the face subspace from the training samples; then the new face image to be identified is projected into d-dimensional subspace by using our algorithm; finally, the new face image is identified by a nearest neighbor classifier. We implemented all the algorithms in Matlab 7.04. The codes as well as the databases in Matlab format can be downloaded on the web.[3]

A. Face Representation Using O-laplacianfaces

In this subsection, we compare the four algorithms for face representation, i.e., Eigenface, Fisherface, Laplacianface, and O-Laplacianface.

[1]http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[2]http://www.uk.research.att.com/facedatabase.html

[3]http://www.ews.uiuc.edu/ dengcai2/Data/data.html

Fig. 2. First six Eigenfaces, Fisherfaces, Laplacianfaces, and O-Laplacianfaces calculated from the face images in the ORL database. (a) Eigenfaces. (b) Fisherfaces. (c) Laplacianfaces. (d) O-Laplacianfaces.

TABLE I
PERFORMANCE COMPARISONS ON THE YALE DATABASE

Method	2 Train	3 Train	4 Train	5 Train
Baseline	56.5%	51.1%	47.8%	45.6%
Eigenfaces	56.5%(29)	51.1%(44)	47.8%(58)	45.2%(71)
Fisherfaces	54.3%(9)	35.5%(13)	27.3%(14)	22.5%(14)
Laplacianfaces	**43.5%(14)**	31.5%(14)	25.4%(14)	21.7%(14)
O-Laplacianfaces	44.3%(14)	**29.9%(14)**	**22.7%(15)**	**17.9%(14)**

For each of them, the basis vectors can be thought of as the basis images and any other image is a linear combination of these basis images. It would be interesting to see how these basis vectors look like in the image domain.

Using the ORL face database, we present the first six O-Laplacianfaces in Fig. 2, together with Eigenfaces, Fisherfaces, and Laplacianfaces.

B. Yale Database

The Yale face database was constructed at the Yale Center for Computational Vision and Control. It contains 165 gray scale images of 15 individuals. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). A random subset with $l(= 2, 3, 4, 5)$ images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given l, we average the results over 20 random splits. Note that, for LDA, there are at most $c-1$ nonzero generalized eigenvalues and, so, an upper bound on the dimension of the reduced space is $c-1$, where c is the number of individuals [2]. In general, the performance of all these methods varies with the number of dimensions. We show the best results and the optimal dimensionality obtained by Eigenface, Fisherface, Laplacianface, O-Laplacianface, and baseline methods in Table I. For the baseline method, the recognition is simply performed in the original 1024-dimensional image space without any dimensionality reduction.

As can be seen, our algorithm performed the best. The Laplacianfaces and Fisherfaces methods performed comparatively to our algorithm, while Eigenfaces performed poorly. Fig. 3 shows the plots of error rate versus dimensionality reduction. It is worthwhile to note that in the cases where only two training samples are available, Fisherfaces method works even worse than baseline and Eigenfaces method. This result is consistent with the observation in [12] that Eigenface method can outperform Fisherface method when the training set is small.

C. ORL Database

The ORL face database is used for this test. It contains 400 images of 40 individuals. Some images were captured at different times and have different variations including expression (open or closed eyes, smiling or nonsmiling) and facial details (glasses or no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 degrees. A random subset with $l(= 2, 3, 4, 5)$ images per individual was taken with labels to form the training set. The rest of the database was considered to be the testing set. For each given l, we average the results over 20 random splits. The experimental protocol is the same as before. The recognition results are shown in Table II and Fig. 4. Our O-Laplacianface method outperformed all the other methods.

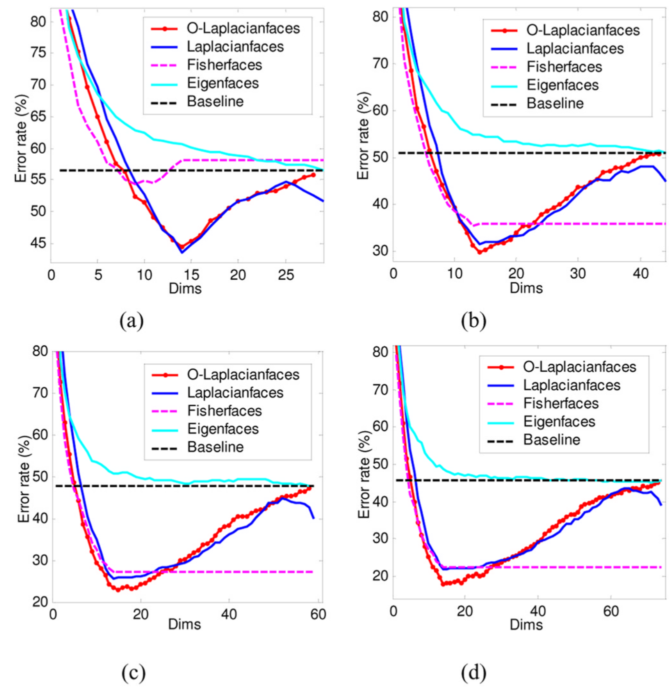

Fig. 3. Error rate versus reduced dimensionality on Yale database: (a) 2 Train; (b) 3 Train; (c) 4 Train; (d) 5 Train.

TABLE II
PERFORMANCE COMPARISONS ON THE ORL DATABASE

Method	2 Train	3 Train	4 Train	5 Train
Baseline	33.8%	24.6%	18.0%	14.1%
Eigenfaces	33.7%(78)	24.6%(119)	18.0%(159)	14.1%(199)
Fisherfaces	28.9%(22)	15.8%(39)	10.5%(39)	7.75%(39)
Laplacianfaces	23.9%(39)	13.4%(39)	9.58%(39)	6.85%(40)
O-Laplacianfaces	**20.4%(40)**	**11.4%(39)**	**5.92%(48)**	**3.65%(59)**

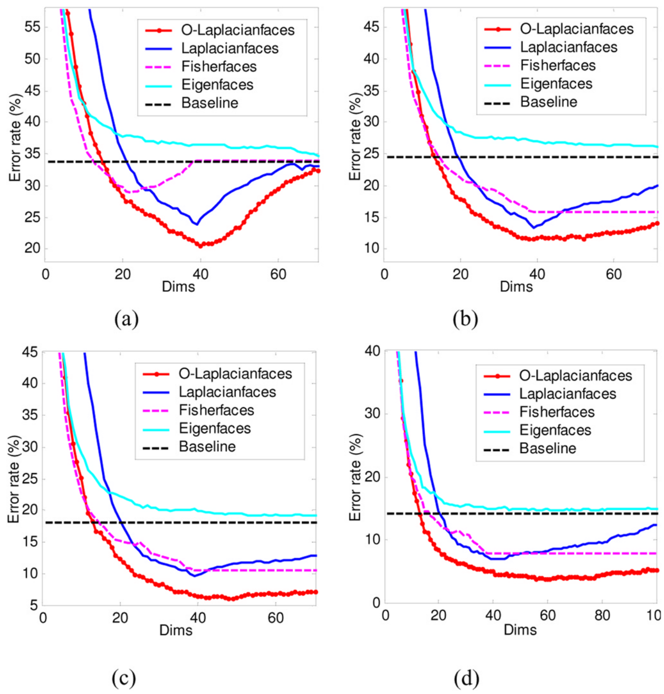

Fig. 4. Error rate versus reduced dimensionality on ORL database: (a) 2 Train; (b) 3 Train; (c) 4 Train; (d) 5 Train.

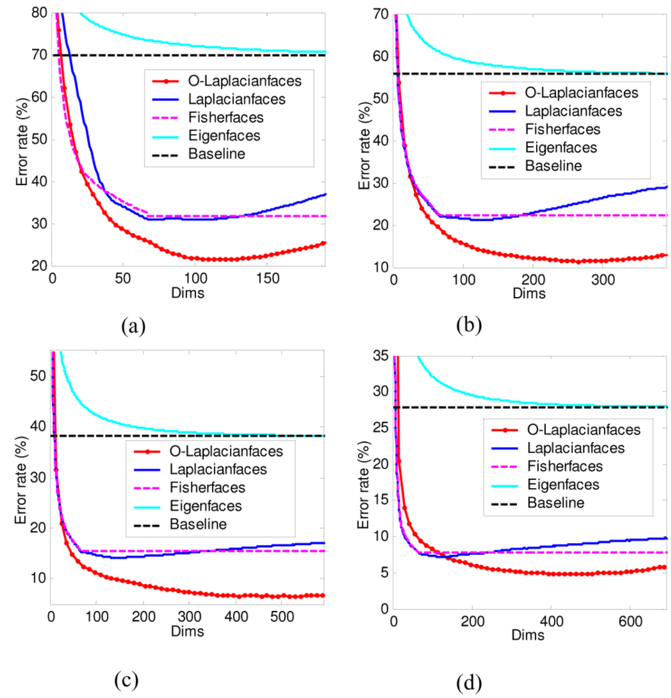

Fig. 5. Error rate versus reduced dimensionality on PIE database: (a) 5 Train; (b) 10 Train; (c) 20 Train; (d) 30 Train.

TABLE III
PERFORMANCE COMPARISONS ON THE PIE DATABASE

Method	5 Train	10 Train	20 Train	30 Train
Baseline	69.9%	55.7%	38.2%	27.9%
Eigenfaces	69.9%(338)	55.7%(654)	38.1%(889)	27.9%(990)
Fisherfaces	31.5%(67)	22.4%(67)	15.4%(67)	7.77%(67)
Laplacianfaces	30.8%(67)	21.1%(134)	14.1%(146)	7.13%(131)
O-Laplacianfaces	**21.4%(108)**	**11.4%(265)**	**6.51%(493)**	**4.83%(423)**

D. PIE Database

The CMU PIE face database contains 68 individuals with 41 368 face images as a whole. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination, and expression. We choose the five near frontal poses (C05, C07, C09, C27, C29) and use all the images under different illuminations, lighting and expressions which leaves us 170 near frontal face images for each individual. A random subset with $l(= 5, 10, 20, 30)$ images per individual was taken with labels to form the training set, and the rest of the database was considered to be the testing set. For each given l, we average the results over 20 random splits. Table III shows the recognition results.

As can be seen, our method performed significantly better than the other methods. The Fisherface and Laplacianface methods performed comparably to each other. The Eigenface method performed the worst. Fig. 5 shows a plot of error rate versus dimensionality reduction.

E. Discussion

We summarize the experiments as follows.

1) Our proposed O-Laplacianface consistently outperforms the Eigenface, Fisherface, and Laplacianface methods.
2) The Fisherface, Laplacianface, and O-Laplacianface methods all outperform the baseline method. Eigenface fails to obtain any improvement. This is probably because it does not encode discriminative information.
3) The low dimensionality of the face subspace obtained in our experiments show that dimensionality reduction is indeed necessary as a preprocessing for face recognition.

VI. CONCLUSION AND FUTURE WORK

We have proposed a new algorithm for face representation and recognition, called orthogonal Laplacianfaces. As shown in our experiment results, orthogonal Laplacianfaces can have more discriminative power than Laplacianfaces.

Several questions remain unclear and will be investigated in our future work.

1) In most previous works on face analysis, it is assumed that the data space is connected. Correspondingly, the data space has an intrinsic dimensionality. However, this might not be the case for real world data. Specifically, the face manifolds pertaining to different individuals may have different geometrical properties, e.g., dimensionality. The data space can be disconnected and different components (individual manifold) can have different dimensionality. It remains unclear how often such a case may occur and how to deal with it.
2) Orthogonal Laplacianfaces is linear, but it can be also performed in reproducing kernel Hilbert space which gives rise to nonlinear maps. The performance of OLPP in reproducing kernel Hilbert space need to be further examined.

REFERENCES

[1] A. U. Batur and M. H. Hayes, “Linear subspace for illumination robust face recognition,” presented at the IEEE Conf. Computer Vision and Pattern Recognition, 2001.
[2] P. N. Belhumeur, J. P. Hepanha, and D. J. 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.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2001, pp. 585–591.
[4] F. R. K. Chung, *Spectral Graph Theory,* 1997, AMS, Regional Conf. Ser. Math.
[5] 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, Jun. 1988.
[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Hoboken, NJ: Wiley-Interscience, 2000.
[7] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[8] R. Gross, J. Shi, and J. Cohn, "Where to go with face recognition," in *Third Workshop on Empirical Evaluation Methods in Computer Vision*, Kauai, HI, Dec. 2001.
[9] 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.
[10] X. He and P. Niyogi, "Locality preserving projections," in *Advances in Neural Information Processing Systems 16.*. Cambridge, MA: MIT Press, 2003.
[11] Q. Liu, R. Huang, H. Lu, and S. Ma, "Face recognition using Kernel based fisher discriminant analysis," presented at the 5th Int. Conf. Automatic Face and Gesture Recognition, Washington, DC, May 2002.
[12] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 228–233, Feb. 2001.
[13] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.
[14] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *Int. J. Comput. Vis.*, vol. 14, 1995.
[15] P. J. Phillips, "Support vector machines applied to face recognition," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 1998, pp. 803–809.
[16] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[17] T. Shakunaga and K. Shigenari, "Decomposed eigenface for face recognition under various lighting conditions," presented at the IEEE Conf. Computer Vision and Pattern Recognition, Dec. 2001.
[18] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illuminlation, and expression database," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1615–1618, Dec. 2003.
[19] J. Tenenbaum, V. d. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
[20] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
[21] M. Turk and A. P. Pentland, "Face recognition using eigenfaces," presented at the IEEE Conf. Computer Vision and Pattern Recognition, Maui, HI, 1991.

Camera Calibration Using Symmetric Objects

Xiaochun Cao and Hassan Foroosh, *Senior Member, IEEE*

***Abstract*—This paper proposes a novel method for camera calibration using images of a mirror symmetric object. Assuming unit aspect ratio and zero skew, we show that interimage homographies can be expressed as a function of only the principal point. By minimizing symmetric transfer errors, we thus obtain an accurate solution for the camera parameters. We also extend our approach to a calibration technique using images of a 1-D object with a fixed pivoting point. Unlike existing methods that rely on orthogonality or pole-polar relationship, our approach utilizes new interimage constraints and does not require knowledge of the 3-D coordinates of feature points. To demonstrate the effectiveness of the approach, we present results for both synthetic and real images.**

***Index Terms*—Camera calibration, homography, stereo reconstruction.**

I. Introduction

Traditional camera calibration methods require a calibration object with a fixed 3-D geometry [1]. Recently, more flexible plane-based calibration methods [2]–[4] are proposed, which use a planar point pattern shown at a few different orientations. Some recent methods in this category use also nonplanar shapes [5]–[7]. Zhang [8] has also presented a method for calibration using 1-D objects. Another branch of study, referred to as self-calibration [9], does not use a calibration object. Their aim is to provide further flexibility by not requiring a prior knowledge of the 3-D to 2-D correspondences. Various methods [10]–[13] have been proposed in this category that rely on scene or motion constraints, most of which require good initialization, and multiple views.

The approach described herein needs a calibration object, but it does not require the knowledge of 3-D coordinates of the object. Different from existing methods, our approach extends the current state-of-the-art calibration methods to situations where only one vanishing point and no vanishing line are known per view. The proposed technique requires the camera to observe a symmetric object only at a few (at least two) different orientations. We will show that such configuration provides sufficient information to solve the problem using only interimage homographies. Given a configuration described shortly, our method can also be applied to 1-D objects [14], which is useful in calibration of a large number of cameras [8].

II. Preliminaries

A homogeneous 3-D point $\tilde{\mathbf{M}} \sim [X \quad Y \quad Z \quad 1]^T$ and its corresponding homogeneous image projection $\tilde{\mathbf{m}}$ in a pinhole camera are related via a 3×4 projection matrix $\mathbf{P}$. More comprehensive imaging models including radial distortion [15] are out of the scope of this paper. When the world points are coplanar (e.g., without loss of generality in the plane $Z = 0$), we get

$$\begin{aligned} \tilde{\mathbf{m}} &\sim \underbrace{\mathbf{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3 \quad \mathbf{t}]}_{\mathbf{P}}\tilde{\mathbf{M}} \\ &= \underbrace{\begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}]}_{\mathbf{H}_w} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \end{aligned} \quad (1)$$

Manuscript received October 7, 2005; revised May 28, 2006. This work was supported by the National Science Foundation (NSF) under Grant IIS-0644280. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Luca Lucchese.

The authors are with the Computational Imaging Lab, School of EECS, University of Central Florida, Orlando, FL 32816-2362 USA (e-mail: xccao@cs.ucf.edu; foroosh@cs.ucf.edu).

Color versions of Figs. 2–7 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.881940