SCIENCE CHINA
Information Sciences

• RESEARCH PAPERS •

December 2010 Vol. 53 No. 12: 2415–2428
doi: 10.1007/s11432-010-4099-1

# Laplacian smoothing transform for face recognition

GU SuiCheng, TAN Ying* & HE XinGui

*Key Laboratory of Machine Perception (MOE); Department of Machine Intelligence, School of Electronics Engineering and Computer Science; Peking University, Beijing 100871, China*

Received March 16, 2009; accepted April 1, 2010

**Abstract** In this paper, we investigate how to extract the lowest frequency features from an image. A novel Laplacian smoothing transform (LST) is proposed to transform an image into a sequence, by which low frequency features of an image can be easily extracted for a discriminant learning method for face recognition. Generally, the LST is able to be an efficient dimensionality reduction method for face recognition problems. Extensive experimental results show that the LST method performs better than other pre-processing methods, such as discrete cosine transform (DCT), principal component analysis (PCA) and discrete wavelet transform (DWT), on ORL, Yale and PIE face databases. Under the leave one out strategy, the best performance on the ORL and Yale face databases is 99.75% and 99.4%; however, in this paper, we improve both to 100% with a fast linear feature extraction method for the first time.

**Keywords** Laplacian smoothing transform (LST), face recognition, principal component analysis (PCA), discrete cosine transform (DCT), discrete wavelet transform (DWT), linear discriminant analysis (LDA)

**Citation** Gu S C, Tan Y, He X G. Laplacian smoothing transform for face recognition. Sci China Inf Sci, 2010, 53: 2415–2428, doi: 10.1007/s11432-010-4099-1

## 1 Introduction

In recent years, numerous algorithms have been proposed for face recognition [1, 2]. However, it is still a difficult task for a machine to recognize human faces accurately, especially under variable circumstances such as variations in illumination, pose, facial expression, etc. [3].

Subspace learning based face recognition methods have attracted much interest. The most widely used subspace learning approach is principal component analysis (PCA). Turk and Pentland [4, 5] used PCA to describe face images in terms of a set of basis functions, or "eigenfaces". However, PCA cannot catch the discriminant information of the samples efficiently. Usually, it was employed to preprocess the raw images for discriminant learning methods. Another well-known approach is the Fisherface in which linear discriminant analysis (LDA) [6] is employed after the PCA is used, by removing the first three principal components [7].

After 2000, manifold based methods are proposed to preserve the local information and obtain a new linear subspace. Some popular ones include locality preserving projection (LPP) [8], neighborhood preserving embedding (NPE) [9], marginal fisher analysis (MFA) [10] and local discriminant embedding (LDE) [11], etc. The LPP and NPE are the projection extension of the local linear embedding (LLE) [12] and Laplacian eigenmaps [13], respectively.

*Corresponding author (email: ytan@pku.edu.cn)

However, the statistical methods and the manifold methods suffer from at least two disadvantages:

- The computational requirements of these approaches are greatly related to the dimensionality of original data and the number of training samples. We have to cropped the original images into small size before training by these techniques, which means much useful information might lost.
- These approaches just regard each pixel on an image to be independent, while a pixel on an image is usually highly related with its neighbors.

In last decade, the discrete cosine transform (DCT) was employed to transform an image into its frequency domain [14–17] for face recognition. It has several advantages over the PCA. First, the DCT is data independent. Second, it can be implemented by a fast algorithm. More recently, many studies shown that the statistical methods, such as LDA, on the DCT frequency domain would improve the recognition rates [18]. Er [19] implemented a high-speed RBF networks in which the LDA is employed on the frequency domain, by removing the first three DCT coefficients [19]. Similar as DCT, the discrete wavelet transform (DWT), as a image compact method, was also introduced for face recognition[20, 21].

Both the DCT and the DWT aim to extract the low frequency smooth features of an image to improve the recognition performances. Wang et al. [22] presented an image Euclidean distance (IMED) and indicated that smoothing noiseless images can increase the recognition rate. Cai et al. [23] introduced a spatially smooth subspace learning (SSSL) model using a Laplacian penalty to constrain the coefficients to be spatially smooth. By using the SSSL, the recognition rates can be improved. However, the IMED and SSSL methods need to crop the original images into small size to reduce the computational complexity.

In this paper, instead of the Laplacian smoothing penalty, a new dimensionality reduction of an image, Laplacian smoothing transform (LST), is deduced from a Laplacian matrix which is derived from the Laplacian smoothing penalty. Unlike the currently frequency transform approaches, such as DCT and DWT, the LST is deduced in an optimal way.

The rest of this paper is organized as follows. In section 2, the related work of Laplacian smoothing is presented. In section 3, the proposed Laplacian smoothing transform is deduced. Section 4 presents the experimental results for face recognition. Finally, a conclusion is given in section 5.

## 2 Laplacian smoothing penalty

### 2.1 Laplacian smoothing

Let $f$ be a function defined on a region of interest, $\mathit{\Omega} \subset \Re^d$. The Laplacian operator $\mathcal{L}$ is defined as follows:

$$\mathcal{L}f(t) = \sum_{j=1}^{d} \frac{\partial^2 f}{\partial t_j^2}. \tag{1}$$

The Laplacian penalty functional, denoted by $\mathcal{J}$, is defined as

$$\mathcal{J}(f) = \int_{\mathit{\Omega}} [\mathcal{L}f]^2 dt. \tag{2}$$

Intuitively, $\mathcal{J}(f)$ measures the smoothness of the function $f$ over the region $\mathit{\Omega}$. In this paper, our primary interest is image, which is the region $\mathit{\Omega}$.

It is worth while to point out that the LST is totally different from Laplacian Eigenmaps and LPP, since LST focuses on pixels in an image while the later two focus on the sample points in sample space.

### 2.2 Discrete Laplacian smoothing

For conventional use, we first define how to represent a image matrix in a column vector.

**Remark.** A vector $v$ with length $MN$ can be written as $v(x * N + y)$, where $x = 0, 1, \ldots, M-1$, $y = 0, 1, \ldots, N-1$. Furthermore, we can alternately write

$$v(\lceil x, y \rceil) = v(x * N + y), \tag{3}$$

where the operator $\lceil \cdot, \cdot \rceil$ is defined as

$$\lceil x, y \rceil = x * N + y. \tag{4}$$

In such a way, a digital image $f(x, y)$ can be easily regarded as a column vector $f(\lceil x, y \rceil)$.

An $MN \times MN$ matrix $L_{MN}$ can be written as $L(x * N + y, x' * N + y')$, where $x, x' = 0, 1, \ldots, M - 1$, $y, y' = 0, 1, \ldots, N - 1$. As a result, we can alternately write

$$L(\lceil x, y \rceil, \lceil x', y' \rceil) = L(x * N + y, x' * N + y'). \tag{5}$$

Thus discretized Laplacian regularization functional of an image $f(x, y)$ can be revised as

$$\mathcal{J}(f) = \sum_{x=0}^{M-2} \sum_{y=0}^{N-1} [f(\lceil x+1, y \rceil) - f(\lceil x, y \rceil)]^2 + \sum_{x=0}^{M-1} \sum_{y=0}^{N-2} [f(\lceil x, y+1 \rceil) - f(\lceil x, y \rceil)]^2. \tag{6}$$

It is clear that eq. (6) is a quadratic form of $f$. With some simple algebraic formulations, we have

$$\mathcal{J}(f) = f^T L_{MN} f, \tag{7}$$

where $L_{MN} = \mathcal{D} - \mathcal{W}$ is an $MN \times MN$ matrix, called Laplacian matrix of an $M \times N$ image and

$$\mathcal{W}(\lceil x, y \rceil, \lceil x', y' \rceil) = \begin{cases} 1, & \text{if } |x - x'| + |y - y'| = 1, \\ 0, & \text{if } |x - x'| + |y - y'| \neq 1. \end{cases} \tag{8}$$

$\mathcal{D}$ is a diagonal matrix whose entries are column (or row, since $\mathcal{W}$ is symmetric) sums of $\mathcal{W}$,

$$\mathcal{D}(\lceil x, y \rceil, \lceil x, y \rceil) = \sum_{\lceil x, y \rceil} \mathcal{W}(\lceil x, y \rceil, \lceil x', y' \rceil). \tag{9}$$

If $|x-x'|+|y-y'| = 1$, then points $(x, y)$ and $(x', y')$ are neighbors. Figure 1 shows a 4-neighbor structure used in this paper. The following is an example of Laplacian matrix $L_{3,2}$.

$$L_{3,2} = \begin{pmatrix} 2 & -1 & 0 & -1 & 0 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 \\ -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & -1 & 0 & -1 & 3 & -1 \\ 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix}.$$

We have some properties of the Laplacian matrix $L_{MN}$, as follows:

- $L_{MN}$ is a sparse matrix with $5MN - 2M - 2N$ non zero elements.
- According to eqs. (6) and (7), $L_{MN}$ is semi-positive matrix with a zero minimal eigenvalue, and its rank is $MN - 1$.
- The eigenvector $e_0$ of $L_{MN}$ corresponding to the zero minimum eigenvalue is with all equal elements.

### 2.3 Spatially smooth subspace learning (SSSL)

The SSSL method introduces a regularized subspace learning model using a Laplacian penalty to constrain the coefficients to be spatially smooth. Suppose we have $m$ face images with size of $M \times N$. Let $\{f_i\}_{i=1}^m$ denote their vector representations and $F = [f_1, \ldots, f_m]$. Let $W$ be a symmetric $m \times m$ matrix with $W_{ij}$ having the weight of the edge joining vertices $f_i$ and $f_j$. $D$ is a diagonal matrix whose entries are column (or row, since $W$ is symmetric) sums of $W$, $D_{ii} = \sum_j W_{ij}$.

Given a pre-defined graph structure with weight matrix $W$, the SSSL approach is to maximize

$$\frac{a^{\mathrm{T}} F W F^{\mathrm{T}} a}{(1 - \alpha) a^{\mathrm{T}} F D F^{\mathrm{T}} a + \alpha \mathcal{J}(a)}, \tag{10}$$

where parameter $0 \leqslant \alpha \leqslant 1$ controls the smoothness of the estimator.

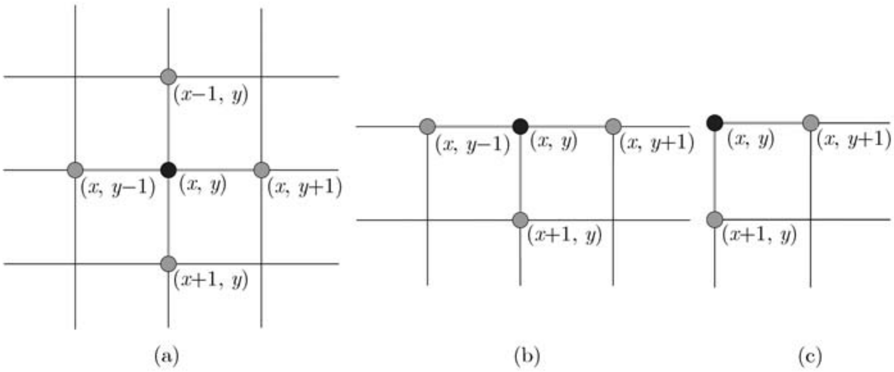

**Figure 1** 4-neighbor structure. An inner point $(i, j)$ has 4 neighbors, an edge point has 3 neighbors, a corner point has 2 neighbors.

For details about the SSSL, please refer to [23]. The experimental results demonstrated that the SSSL method can significantly improve the recognition rate on the ORL, Yale and PIE databases. However, it still suffers from the problem of computational complexity.

### 2.4 The isotropic smoothing function

A method called isotropic smoothing [24, 25] aims to obtain an illumination function $h(x, y)$ that is similar to the original image $f(x, y)$, but contains a smoothing constraint. The objective function can be constructed by minimizing the following cost function:

$$\Psi(h) = \|h - f\|^2 + c\mathcal{J}(h), \tag{11}$$

where parameter $c > 0$ controls the relative importance of the smoothness constraint. This equation should be solved using multi-grid methods [26].

## 3 Laplacian smoothing transform of an image

In this section, the proposed Laplacian smoothing transform (LST) is introduced. The partial Laplacian smoothing transform (PLST) is presented for dimensionality reduction of an image. We also show that the LST can be easily combined with the discriminant learning methods.

### 3.1 Laplacian smoothing transform (LST)

**Lemma 1.** Laplacian matrix $L_{MN}$ is defined in subsection 2.2. $MN$ eigenvalues of $L_{MN}$ are

$$0 = \lambda_0 < \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_{MN-1}, \tag{12}$$

whose corresponding eigenvectors are

$$e_0, e_1, e_2, \ldots, e_{MN-1}. \tag{13}$$

Then,

$$\mathcal{J}(e_i) \leqslant \mathcal{J}(e_j),\ \forall i < j. \tag{14}$$

*Proof.* According to eq. (7), one can obtain

$$\mathcal{J}(e_i) = e_i^{\mathrm{T}} L_{MN} e_i = \lambda_i, \tag{15}$$

$$\mathcal{J}(e_j) = e_j^{\mathrm{T}} L_{MN} e_j = \lambda_j. \tag{16}$$

Since $i < j$, then $\lambda_i \leqslant \lambda_j$, hence $\mathcal{J}(e_i) \leqslant \mathcal{J}(e_j)$.

An eigenvector of the Laplacian matrix $L_{MN}$, e.g., $e(\lceil x, y \rceil)$, can be regarded as an $M \times N$ matrix $e(x, y)$. Figure 2 shows the first 20 eigenvectors of the Laplacian matrix $L_{32,32}$.

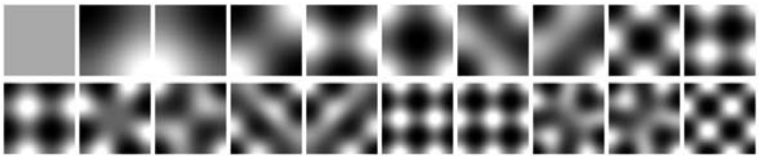

**Figure 2** Eigenvectors corresponding to the first 20 minimal eigenvalues of the Laplacian matrix $L_{32,32}$.

From Lemma 1, we know that the eigenvector according to smaller eigenvalue of the Laplacian matrix $L_{MN}$ is smoother, since $\mathcal{J}(\cdot)$ can measure the smoothness of an image. Therefore, we can decompose an image $f(x,y)$ into its frequency domain by projecting an image onto the eigenvectors of the Laplacian matrix:

**Definition 1** (Laplacian smoothing transform (LST))**.** Let the $MN \times MN$ matrix $\mathit{\Gamma} = (e_0, e_1, e_2, \ldots, e_{MN-1})$, where $\{e_k\}_{k=0}^{MN-1}$ are eigenvectors of the Laplacian matrix $L_{MN}$ as in eq. (13). Then, Laplacian smoothing transform (LST) of an $M \times N$ image $f(\lceil x, y \rceil)$ can be defined as

$$LST(f) = \mathit{\Gamma}^{\mathrm{T}} f, \tag{17}$$

where symbol 'T' denotes the transpose of a matrix.

**Definition 2** (ILST)**.** Assume the column vector $g$ is the Laplacian smoothing transform of the image $f$, i.e., $g = LST(f)$. Then the inverse Laplacian smoothing transform (ILST) of $g$ is given by

$$f^* = ILST(g) = \mathit{\Gamma} g. \tag{18}$$

It is obvious to have $f^* = \mathit{\Gamma}\mathit{\Gamma}^{\mathrm{T}} f = f$. According to eqs. (17) and (18), we can transform any image $f$ into its LST frequency domain easily and accurately and vice versa.

However, if the image size $M \times N$ changes we should compute different basis for the new image. Therefore, the LST is more efficient for the images with same size.

### 3.2 Partial Laplacian smoothing transform (PLST)

As a matter of fact, the human visual system is more sensitive to variations in the low-frequency band. The detailed discussions between frequency and recognition can be found in [19]. In order to obtain low-frequency features of an $M \times N$ image $f(\lceil x, y \rceil)$, we just select the first $k$ elements of $g$, where $g = LST(f)$ as in eq. (17), which can be directly calculated in the following steps:

1. Construct the Laplacian matrix $L_{MN}$;
2. Compute $k$ minimal eigenvalues and their corresponding eigenvectors $\{e_i\}_{i=0}^{k-1}$;
3. Let $\mathit{\Gamma}_k = (e_0, e_1, \ldots, e_{k-1})$, then

$$g = \mathit{\Gamma}_k^{\mathrm{T}} f \tag{19}$$

is the feature vector of $k$ low-frequency LST coefficients of image $f$.

**Definition 3** (Low pass filter)**.** According to eqs. (18) and (19), we can partially reconstruct $f$ with low frequency features by

$$f^* = \mathit{\Gamma}_k g = \mathit{\Gamma}_k \mathit{\Gamma}_k^{\mathrm{T}} f, \tag{20}$$

where $\mathit{\Gamma}_k \mathit{\Gamma}_k^{\mathrm{T}}$ is called as a low pass filter of $f$.

Figure 3 shows an example of the low pass filter of the LST.

### 3.3 Solution of the isotropic smoothing function

**Lemma 2.** $\{e_k\}_{k=0}^{MN-1}$ and scalars $\{\lambda_k\}_{k=0}^{MN-1}$ are the eigenvectors and eigenvalues, respectively, of the Laplacian matrix $L_{MN}$. Given an $M \times N$ image $f(\lceil x, y \rceil)$, which can be expanded as

$$f = \sum_{k=0}^{MN-1} a_k e_k. \tag{21}$$

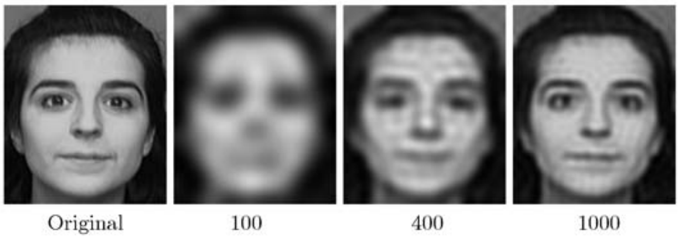

**Figure 3** An example of the low past filter of the LST. The left is an original image with size of 112 × 92. The other three are the images reconstructed with $k = 100, 400$ and 1000 coefficients of LST.

If $h(\lceil x, y\rceil)$ is the solution of eq. (11), then

$$h = \sum_{k=0}^{MN-1} \frac{1}{1 + c\lambda_k} a_k e_k. \tag{22}$$

The proof of Lemma 2 can be found in Appendix A. Therefore, the isotropic smoothing function can be directly solved by our proposed LST as in eq. (22).

Since

$$\lambda_0 < \lambda_1 \leqslant \cdots \leqslant \lambda_{MN-1}, \tag{23}$$

one can obtain

$$\frac{1}{1 + c\lambda_0} > \frac{1}{1 + c\lambda_1} \geqslant \cdots \geqslant \frac{1}{1 + c\lambda_{MN-1}}. \tag{24}$$

From eqs. (21) and (22), we can see that this isotropic smoothing method gives the low-frequency features of the original image with greater multiplier to obtain a smoother function. Alternately, the LST in this paper cuts off the high frequency features directly. Therefore, the LST not only discards high frequency noises but also reduce the computational complexity.

### 3.4 Combining with learning algorithms

Given $m$ images $F = (f_1, f_2, \ldots, f_m)$, each with size of $M \times N$, we first compute $\Gamma_k$ as in eq. (19), each sample $f_i$ can be transformed to its subspace $g_i = \Gamma_k^{\mathrm{T}} f_i$, $F$ is then represented by $G = \Gamma_k^{\mathrm{T}} F = [g_1, \ldots, g_m]$. Different from the SSSL approach as in subsection 2.3, the maximization problem is to find

$$u* = \arg\max \frac{u^{\mathrm{T}} G W G^{\mathrm{T}} u}{u^{\mathrm{T}} G D G^{\mathrm{T}} u}. \tag{25}$$

Let $U_l = (u_1, u_2, \ldots, u_l)$, where $u_i$ is the $i$th eigenvector of the problem. Matrix $U_l$ is with size of $k \times l$. Then, the transformation matrix is $V = \Gamma_k U_l$ with size of $MN \times l$. Given a new image $f$ with size of $M \times N$, the final discriminant feature of $f$ is $V^{\mathrm{T}} f$.

Figure 4 shows the one-step and two-step linear dimensionality reduction framework.

**Lemma 3.** Suppose $v$ is a column vector from matrix $V$, then $\mathcal{J}(v) \leqslant \lambda_k$, where $\lambda_k$ is the $k$th minimal eigenvalue of the Laplacian matrix, defined in eq. (12).

Proof of Lemma 3 is given in Appendix B. Ten basis images $(v_1, \ldots, v_{10})$ computed by "LST+LDA" are shown in Figure 5. It can be seen from Lemma 3 that each axis $v$ is spatially smooth and it can be robust to variations of poses and expressions of faces.

### 3.5 Computational complexity

For the LST, the computational complexity mostly lies in the computational amount of the $k$ minimal eigenvectors and corresponding eigenvectors of the Laplacian matrix $L_{MN}$. Fortunately, $L_{MN}$ is a sparse matrix, it can be computed fast as LLE and Laplacian eigenmaps. On the other hand, it only needs to be calculated only once for training. For example, to calculate first 200 minimal eigenvalues of $L_{100,100}$, only 10 s is needed.

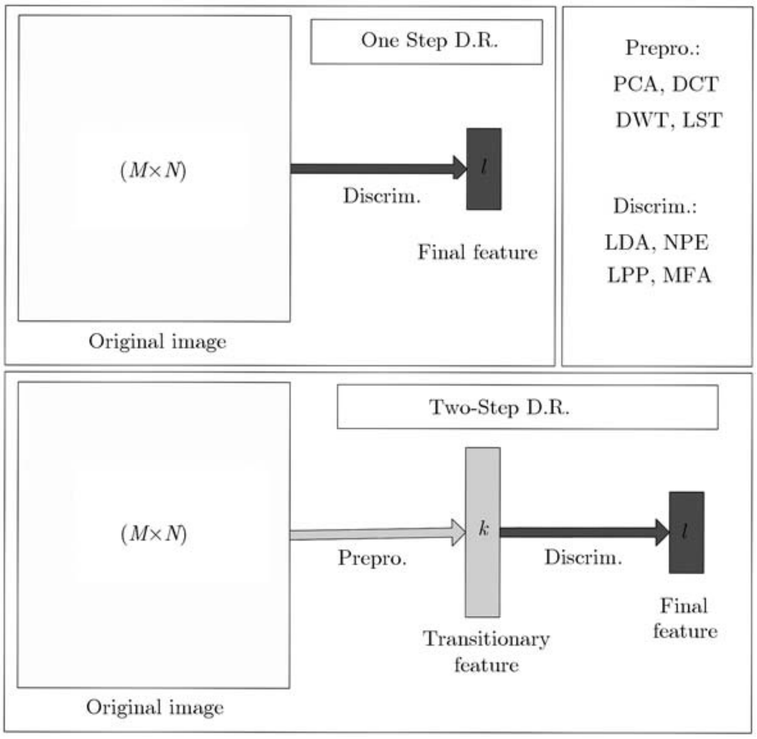

**Figure 4** Dimensionality reduction framework.

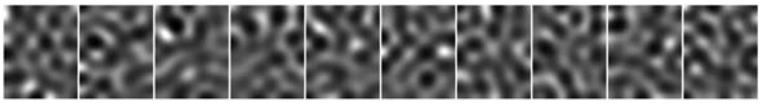

**Figure 5** First ten basis images $v$ computed by the two step method “LST+LDA” on the Yale database.

All the LDA, NPE, LPP, MFA, SSSL methods suffer from the problem of computational complexity. If the images are with size of $M \times N$, these methods need to compute the eigenvalues of an $MN \times MN$ full matrix. Even the images are not very large, such as $64 \times 64$, it costs more than 2 h to compute the eigenvalues of a full matrix with size of $4096 \times 4096$. Therefore, it is not practical to implement these methods on the original images directly (see One Step D.R. in Figure 4).

In order to accelerate the computing, we should reduce the dimensionality of the images before. A direct way is to cut or compact the images into small size. But, much useful information would be lost during this pre-processing. Alternately, we can employ some dimensionality reduction methods for pre-processing, such as DCT, DWT, PCA, LST, etc. While the image samples are reduced to much smaller size, the LDA, NPE, LPP, MFA methods can be computed much faster (see Two-Step D.R. in Figure 4).

## 4 Experiments and discussions

### 4.1 Experimental setup

In order to evaluate the proposed face recognition system, our experiments are conducted on four benchmark face databases: 1) The Olivetti Research Laboratory (ORL) database [27]; 2) the Yale database [7]; 3) the PIE database [28]; 4) the Feret database. Properties of the first three data sets are given in Tables 1 and 2. Each face image vector was normalized to unit before use.

In order to fairly evaluate different feature extraction methods, the nearest neighbor classifier is used for classification task. To obtain the nearest neighbor, the Euclidean distance measure is used. Our experimental platform is a Pentium IV 1.73G personal computer under windows xp operational system

**Table 1** Properties of data sets

| Datasets | Samples | Classes | Size | Sample figure |
|---|---|---|---|---|
| ORL | 400 | 40 | 112×92 | Figure 6 |
| Yale | 165 | 15 | 200×160 | Figure 7 |
| PIE | 11154 | 68 | 64×64 | Figure 8 |

**Table 2** Properties of the dimensional reduction methods

| Methods | Objective | Data relevance | speed |
|---|---|---|---|
| DWT | frequency transform | data independent | fast |
| DCT | frequency transform | data independent | fast |
| LST | frequency transform | data independent | fast |
| PCA | statistical | unsupervised | medium |
| LDA | statistical learning | supervised | slow |
| LPP | manifold learning | supervised | slow |
| NPE | | | |
| MFA | manifold learning | supervised | slow |

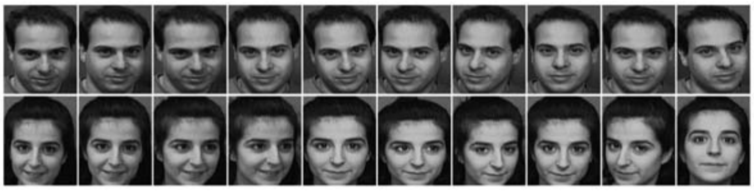

**Figure 6** Two subjects of the ORL database. For some subjects, the images were taken at different times. The images were taken with a tolerance for some tilting and rotation of the face of up to 20 degrees.

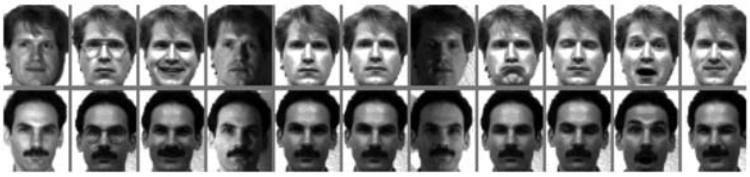

**Figure 7** Two subjects of Yale face database. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

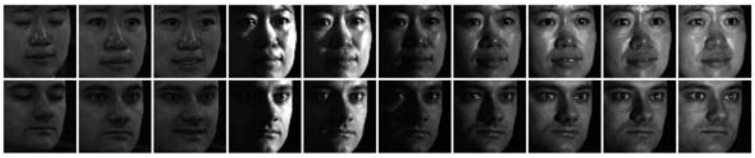

**Figure 8** Two subjects of PIE face database. All the images under different illuminations and expressions.

and Matlab 7.4. The codes in matlab and an example of LST is available at footnote 1).

Experimental remark:

- DWT: The Haar basis is chosen in this paper.
- DCT: The Zig-zag manner is used to convert DCT coefficient matrix into a one-dimensional vector [16].
- LPP, NPE: The original LPP and NPE are unsupervised, but the revised supervised versions are used in this paper. For details, please refer to this link[2].

[1] http://www.mathworks.com/matlabcentral/fileexchange/23251

[2] http://www.cs.uiuc.edu/homes/dengcai2/Data/data.html

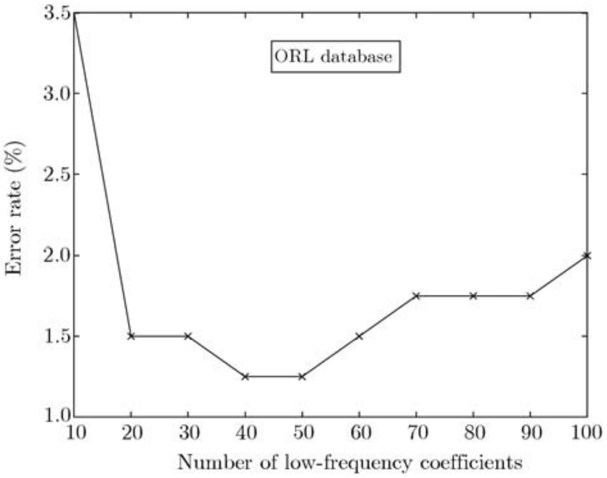

**Figure 9** Error rates of selecting different numbers of low-frequency coefficients of LST under the Leave one out strategy.

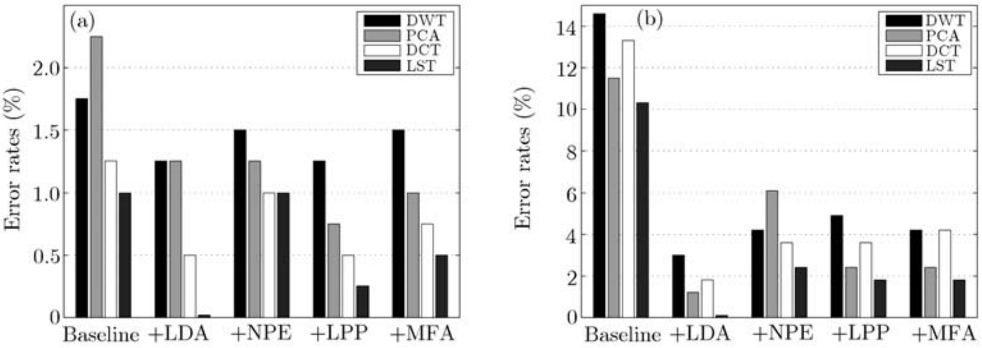

**Figure 10** Comparison on ORL (a) and Yale (b) face databases. The first group of "Baseline" evaluates the four preprocessing methods with no further learning method. The next four groups show the performances of two-step methods.

### 4.2 Number of LST coefficients

Figure 9 shows the error rates of selecting different numbers of low-frequency LST coefficients on the ORL database, respectively. The best result is obtained if 40 low frequency LST coefficients are selected. When we select more than 50 coefficients, the error rate goes up significantly. The result indicates that the high frequency features are not useful for face recognition and only cost much time for computation. Therefore, LST can help to reduce the computational complexity and improve performances by cutting high frequency features.

### 4.3 Comparisons on ORL and Yale face databases using the leave-one-out strategy

For the ORL and the Yale databases, our system is first tested by using the "leave-one-out" strategy which means that 400 experiments and 165 experiments are constructed, respectively, for the ORL and Yale databases. For each experiment, one image is removed from the data set and all the remaining images are used for training. The error rate of this strategy is given as the average error rate of all experiments. Comparisons on the ORL database and the Yale database are shown in Figure 10(a) and (b), respectively. The experimental results of the four preprocessing on ORL and Yale databases are shown in the first group of each figure. And the four preprocessing methods with four further learning methods are shown in the next four groups. The proposed LST method yield best results among those preprocessing approaches on both databases.

Many experimental results have been reported on ORL and Yale face databases. On ORL database, the best result before is 99.75%, by using kernel LDE [11]. On Yale database, the best result before is

**Table 3** Comparison on ORL and Yale database (mean±std-dev%)$^{a)}$

| ORL | G2/P8 | G3/P7 | G4/G6 | G5/P5 |
|---|---|---|---|---|
| Baseline | 33.1±3.4 | 23.4±2.3 | 17.9±2.2 | 13.7±2.4 |
| DWT+LDA | 18.7±3.1 | 10.7±1.9 | 6.5±1.7 | 4.6±1.4 |
| PCA+LDA | 24.5±3.3 | 13.9±1.9 | 8.4±1.9 | 5.7±1.4 |
| DCT+LDA | 16.7±3.2 | 9.6±2.3 | 5.6±1.8 | 3.5±1.4 |
| S-LDA [23] | **14.8±2.2** | **7.7±1.7** | **4.2±1.3** | **2.8±1.3** |
| LST+LDA | 15.5±2.9 | 7.9±1.8 | 5.0±1.6 | 3.1±1.2 |
| Yale | G2/P9 | G3/P8 | G4/P7 | G5/P6 |
| Baseline | 56.6±3.9 | 50.6±4.2 | 47.4±3.9 | 43.8±3.1 |
| DWT+LDA | 29.6±4.2 | 19.7±3.6 | 11.2±2.5 | 9.4±2.3 |
| PCA+LDA | 42.5±4.7 | 31.3±3.7 | 25.5±4.6 | 21.6±3.4 |
| DCT+LDA | 25.4±3.8 | 17.6±3.4 | 9.8±2.3 | 8.6±2.2 |
| S-LDA | 37.5±4.9 | 25.6±4.6 | 19.7±3.3 | 14.9±3.2 |
| LST+LDA | **16.0±3.6** | **9.8±2.8** | **6.1±2.3** | **3.9±1.1** |

a) G=gallery, P=probe. G5/P5 means, for each individual, 5 samples are used for train and other 5 samples are used for test. Baseline: using original images for classification.

**Table 4** Comparison on the PIE database (mean±std-dev%)

| Methods | G5/P165 | G10/P160 | G20/P150 | G50/P120 | G70/P100 | G90/P80 | G130/P40 |
|---|---|---|---|---|---|---|---|
| Baseline | 69.9±0.8 | 55.6±0.9 | 38.2±0.7 | 16.3±0.5 | 10.6±0.4 | 7.2±0.4 | 3.9±0.3 |
| DWT+LDA | 40.6±1.3 | 24.8±0.8 | 14.1±0.6 | 6.1±0.3 | 4.4±0.3 | 3.4±0.2 | 2.5±0.3 |
| PCA+LDA | 37.7±1.2 | 22.3±0.8 | 12.5±0.5 | 5.4±0.2 | 4.1±0.3 | 3.3±0.2 | 2.5±0.2 |
| DCT+LDA | 30.7±1.2 | 17.2±0.8 | 9.7±0.6 | 4.2±0.2 | 3.4±0.3 | 2.7±0.2 | 2.1±0.2 |
| **LST+LDA** | **28.1±1.0** | **13.7±0.6** | **6.8±0.4** | **3.1±0.2** | **2.3±0.2** | **1.9±0.1** | **1.6±0.2** |

**Table 5** Four probe subsets and their evaluation task

| Probe subsets | Evaluation task | Number of images |
|---|---|---|
| Dup1 | Aging of subjects | 722 |
| Dup2 | Aging of subjects | 234 |
| fafb | Facial expression | 1195 |
| fafc | Illumination | 194 |

**Figure 11** Facial images from the fafc subset.

99.4%, by using Fisherface [7]. On both ORL and Yale face databases, we obtain 100% recognition rates by using “LST+LDA” method. Also, the LST+LDA method is much faster than the kernel LDE and fisherface.

## 4.4 Comparisons on the ORL, Yale and PIE databases over 50 random splits

Since the LDA yields the best result in our first two experiments, we only implement the LDA learning algorithm with the four preprocessing methods, such as DWT, PCA, DCT and LST. Tables 3 and 4 show the results of different numbers of training samples. For each Gp/Pq, we average the results over 50 random splits and report the mean as well as the standard deviation. Our proposed LST method obtain the best results on all the three databases among the four preprocessing methods. Only on the ORL database, the S-LDA method outperforms the LST+LDA method slightly. However, the advantage is trivial. On the other hand, the LST+LDA method is much faster than the S-LDA method.

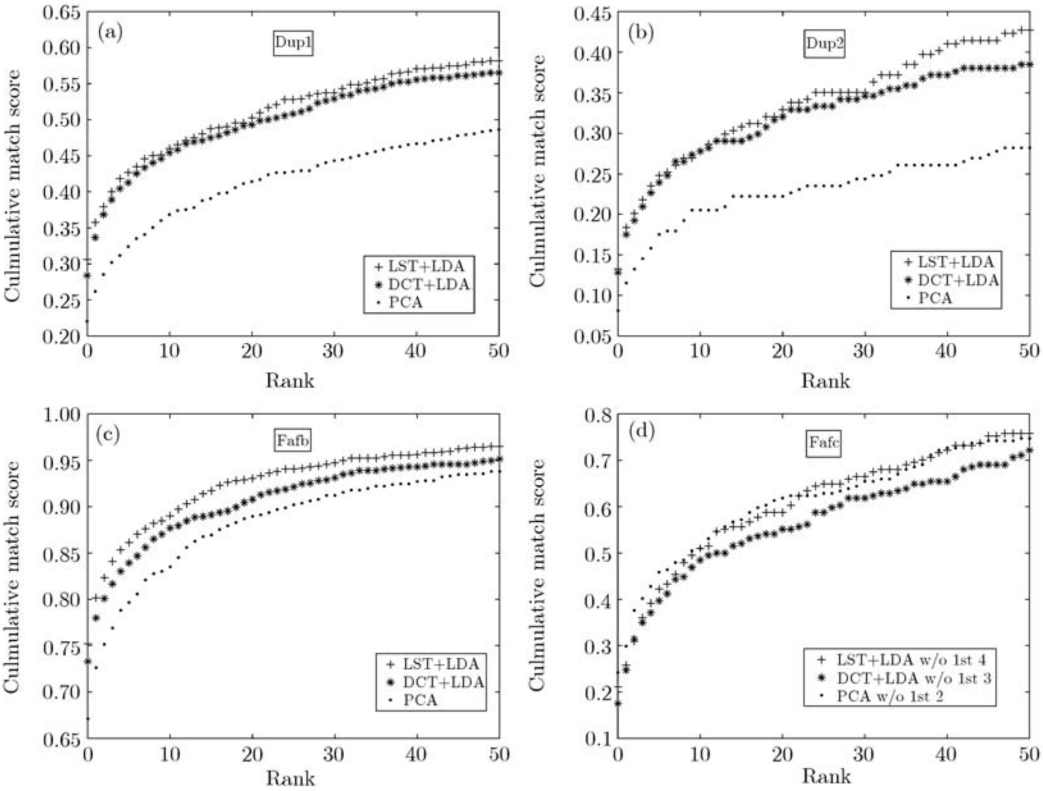

**Figure 12** Cumulative match curves of LST+LDA, DCT+LDA and PCA on the FERET database.

### 4.5 Testing on FERET database

The proposed LST method is also tested on the FERET database which contains more subjects with different variations. The gallery, which has 1196 images, is used for train. Four probe subsets are used for test (see Table 5). Five subjects from the "fafc" probe are shown in Figure 11. We employ the Colorado State University (CSU) Face Identification Evaluation System to evaluate the proposed feature extraction method [29]. We compare our method with the DCT and the PCA methods in the following way.

1. For the "LST+LDA" approach, 100 LST coefficients are obtained. By discarding the first one, 99 coefficients are used and the dimensionality of the feature vectors is 70 after the LDA is applied.

2. For the "DCT+LDA" approach, 100 DCT coefficients are obtained. By discarding the first three (this strategy is suggested in [17]), 97 coefficients are used and the dimensionality of the feature vectors is 70 after the LDA is applied.

3. For the PCA approach, because the gallery has one image from each class, PCA+LDA would be the same as the baseline PCA, so we just use the baseline PCA to obtain 70 coefficients.

To generate the cumulative match curve, the Euclidean distance measure is used. It should be noted that, for the "fafc" probe, we do five experiments, by discarding first 0, 1, 2, 3, 4 of the 70 remained coefficients, respectively, for each of the three methods. For each method, the curve in Figure 12(fafc) shows the best of the five experiments. It can be seen from Figure 12 that the proposed method "LST+LDA" outperforms both the "DCT+LDA" and PCA methods.

### 4.6 Discussions

The two-step subspace learning methods can not only help to reduce the computational complexity but also help to improve the recognition performance. The one step subspace learning methods (LDA, NPE, MFA) only consider the image as a point in the sample space while the images are high related itself.

Consider the four preprocessing methods, PCA, DWT, DCT, LST. The PCA also consider an image as a point in the original space. The DWT and DCT are used for image compaction with given basis.

The LST is induced in an optimal way by the Laplacian smooth matrix. Also, the experimental results show that the LST method is the best one of the four preprocessing methods.

Even though the Manifold methods, such as LPP, NPE, MFA, are newer techniques than LDA, they did not obtain better results than LDA. On the other hand, since the manifold methods have many parameters to tune, they are not easy to be implemented in practice. Table 2 shows the properties of the dimensional reduction methods.

## 5 Conclusions

In this paper, a new method, Laplacian smoothing transform (LST), is proposed for dimensionality reduction of an image. The LST transforms an image into a one-dimensional sequence (frequency domain). Unlike the PCA, the proposed LST is data independent, thus it can be easily dealt with the new input images. Unlike the Laplacian smoothing penalty methods, such as SSSL, LST can cut the high frequency features directly and thus be much faster. Unlike the image compression approaches such as DWT and DCT, LST is deduced in an optimal way, thus it can obtain the lowest frequency features.

By using the a pre-processing method, such as LST, the computational complexity of the discriminant learning methods can be greatly reduced and the recognition rates can be greatly improved. The LST can obtain the higher recognition rates than the other preprocessing methods, such as DCT, DWT and PCA on ORL, Yale, PIE and FERET databases. Especially, on ORL and Yale face databases, just combining the proposed LST with LDA, we obtained the perfect 100% recognition rates of the leave-one-out strategy for the first, though the two databases have been studied for many years by a number of approaches.

**Acknowledgements**

This work was supported by the National High-Tech Research & Development Program of China (Grant No. 2007AA01Z453), and the National Natural Science Foundation of China (Grant Nos. 60875080, 60673020). We thank the anonymous referees for their valuable comments on this paper.

### References

1 Chellappa R, Wilson C L, Sirohey S. Human and machine recognition of faces: A survey. Proc IEEE, 1995, 83: 705–740

2 Zhao W, Chellappa P J R, Rosenfeld A. Face recognition: A literature survey. ACM Comput Surveys, 2003, 35: 399–458

3 Adini Y, Moses Y, Ullman S. Face recognition: the problem of compensating for illumination changes. IEEE Trans Patt Anal Mach Intell, 1997, 19: 721–732

4 Kirby M, Sirovich L. Application of the kl procedure for the characterization of human faces. IEEE Trans Patt Anal Mach Intell, 1990, 12: 103–108

5 Turk M, Pentland A. Eigenfaces for recognition. J Cognit Neurosci, 1991, 3: 71–86

6 Duda R, Hart P. Pattern Classification and Scene Analysis. New York: Wiley, 1973

7 Belhumeur P N, Hespanha J P, Kriegman D J. Eigenfaces versus fisherfaces: Recognition using class specific linear projection. IEEE Trans Patt Anal Mach Intell, 1997, 19: 711–720

8 He X F, Yan S C, Hu Y, et al. Face recognition using Laplacianfaces. IEEE Trans Patt Anal Mach Intell, 2005, 27: 328–340

9 He X F, Cai D, Yan S C, et al. Neighborhood preserving embedding. In: Tenth IEEE International Conference on Computer Vision (ICCV'05), Beijing, China, 2005, 2: 1208–1213

10 Yan S C, Xu D, Zhang B, et al. Graph embedding and extensions: A general framework for dimensionality reduction. IEEE Trans Patt Anal Mach Intelle, 2007, 29: 40–51

11 Chen H T, Chang H W, Liu T L. Local discriminant embedding and its variants. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR05), San Diego, California, 2005, 2: 846–853

12 Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290: 2323–2326

13 Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing System. Cambridge: MIT Press, 2001. 585–591

14 Kohir V V, Desai U B. Face recognition using dct-hmm approach. In: Fourth IEEE Workshop on Applications of Computer Vision (WACV'98), Princeton, New Jersey, 1998. 226

15 Pan Z, Adams R, Bolouri H. Image redundancy reduction for neural network classification using discrete cosine transforms. In: Proc. IEEE INNS ENNS Int. Joint Conf. Neural Networks, Como, Italy, 2000, 3: 149–154

16 Hafed Z M, Levine M N. Face recognition using the discrete cosine transform. Int J Comput Vision, 2001, 43: 167–188

17 Yu M, Yan G, Zhu Q W. New face recognition method based on dwt/dct combined feature selection. In: Proc. 5th International Conf. Machine Learning and Cybernetics, Dalian, China, 2006. 13–16

18 Zhou D, Yang X. Face recognition using enhanced fisher linear discriminant model with facial combined feature. PRICAI, 2004, LNAI 3157: 769–777

19 Er M J, Chen W, Wu S. High-speed face recognition based on discrete cosine transform and rbf neural networks. IEEE Trans Neural Netw, 2005, 16: 679–691

20 Wiskott L, Fellous J M, Kruger N, et al. Face recognition by elastic bunch graph matching. Int Conf Image Process, 1997, 1: 129–132

21 Dai D Q, Yuen P. Wavelet based discriminant analysis for face recognition. Appl Math Comput, 2006, 175: 307–318

22 Wang L, Zhang Y, Feng J. On the euclidean distance of images. IEEE Trans Patt Anal Mach Intell, 2005, 27: 1334–1339

23 Cai D, He X, Hu Y, et al. Learning a spatially smooth subspace for face recognition. In: IEEE Conf. Computer Vision and Pattern Recognition (CVPR07), Minneapolis, MN, USA, 2007. 1–7

24 Short J, Kittler J, Messer K. A comparison of photometric normalization algorithms for face verification. In: Proc. 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR2004), 2004. 254–259

25 Gross R, Brajovic V. An image preprocessing algorithm for illumination invariant face recognition. In: Audio and Video-based Biometric Person Authentication (AVBPA), LNCS 2688, 2003. 10–18

26 Briggs W, Henson V, McCormick S. A Multigrid Tutorial. 2nd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1987

27 Samaria F, Harter A. Parameterisation of a stochastic model for human face identification. In: 2nd IEEE Workshop on Applications of Computer Vision, Sara-sota, Florida, 1994. 138–142

28 Sim T, Baker S, Bsat M. The cmu pose, illumination, and expression database. IEEE Trans Patt Anal Mach Intell, 2003, 25: 1615–1618

29 Beveridge R, Bolme D, Teixerira M, et al. The CSU Face Identification Evaluation System Users Guide: Version 5.0. Fort Collins, CO: Colorado State Univ., May 2003

## Appendix A

### Proof of Lemma 2 in section 3.3

*Proof.* Suppose $h(\lceil x, y\rceil)$ can be expanded by LST as

$$h(\lceil x, y\rceil) = \sum_{k=0}^{MN-1} b_k e_k, \tag{A1}$$

where $b_k$ are the LST coefficients of $h$, then

$$\|h - f\|^2 = \sum_{k=0}^{MN-1} (a_k - b_k)^2, \tag{A2}$$

According to eq. (7), we have

$$\mathcal{J}(h) = h^{\mathrm{T}} L_{MN} h = \left(\sum_{k=0}^{MN-1} b_k e_k\right)^{\mathrm{T}} L_{MN} \left(\sum_{k=0}^{MN-1} b_k e_k\right) = \sum_{k=0}^{MN-1} b_k^2 \lambda_k. \tag{A3}$$

Substitute eqs. (A2) and (A3) into eq. (11), we can obtain

$$\Psi(h) = \sum_{k=0}^{MN-1} (a_k - b_k)^2 + c \sum_{k=0}^{MN-1} b_k^2 \lambda_k, \tag{A4}$$

hence the first derivative of $\Psi(h)$ with respect to $b_k$ is

$$\frac{\partial \Psi(h)}{\partial b_k} = 2(b_k - a_k) + 2c b_k \lambda_k = 0, \tag{A5}$$

therefore

$$b_k = \frac{a_k}{1 + c\lambda_k}. \tag{A6}$$

## Appedix B

### Proof of Lemma 3 in section 3.4

*Proof.* We have $V = (v_1, v_2, \ldots, v_l) = \mathit{\Gamma}_k U$, hence the $i$th column of $V$ is

$$v_i = \mathit{\Gamma}_k u_i. \tag{B1}$$

Then

$$J(v_i) = u_i^{\mathrm{T}} \Gamma_k L_{MN} \Gamma_k u_i. \tag{B2}$$

Since $\mathit{\Gamma}_k L_{MN} \mathit{\Gamma}_k = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$, let $u_i(j)$ denotes the $j$th element of $u_i$, we have

$$J(v_i) = \sum_{j=1}^{k} \lambda_j u_i(j)^2 \leqslant \sum_{j=1}^{k} \lambda_k u_i(j)^2 = \lambda_k. \tag{B3}$$