Locality Preserving CCA with Applications to Data Visualization and Pose Estimation

Tingkai Sun Songcan Chen*

Dept. of Computer Science & Engineering, Nanjing University of Aeronautics & Astronautics

Nanjing, 210016, P.R. China

Abstract - Canonical correlation analysis (CCA) is a major linear subspace approach to dimensionality reduction and has been applied to image processing, pose estimation and other fields. However, it fails to discover or reveal the nonlinear correlation relationship between two sets of features. In contrast, its kernelized nonlinear version, KCCA, can overcome such a shortcoming, but the global kernelization of CCA restrains KCCA itself from effectively discovering the local structure of the data with complex and nonlinear characteristics. Recently, the locality methods, such as locally linear embedding (LLE) and locality preserving projections (LPP), are proposed to discover the low dimensional manifold embedded in the original high dimensional space. Compared to the subspace based methods, these locality methods take into account the local neighborhood structure of the data, and can discover the intrinsic structure of data to a better degree, which benefits to the subsequent computation. Inspired by the locality based methods, in this paper, we incorporate such an idea into CCA and propose locality preserving CCA (LPCCA) to discover the local manifold structure of the data and further apply it to data visualization and pose estimation. In addition, a fast algorithm of LPCCA is proposed for some special cases. The experiments show that LPCCA can both capture the intrinsic structure characteristic of the given data and achieve higher pose estimation accuracy than both CCA and KCCA.

Keywords: Canonical correlation analysis (CCA); Locality preservation; Pose estimation; Data visualization; Dimensionality reduction.

* Corresponding author: Tel: +86-25-84896481 Ext. 12106; Fax:+86-25-84498069; E-mail: s.chen@nuaa.edu.cn (S. Chen) and suntingkai@nuaa.edu.cn (T. Sun)

1. Introduction

As one of the principal subspace approaches to dimensionality reduction, canonical correlation analysis (CCA) [1] aims to find basis vector pairs, ($\boldsymbol{w}_x$, $\boldsymbol{w}_y$), for two sets of mean-normalized variables $\boldsymbol{X}$=[$\boldsymbol{x}_1$,…, $\boldsymbol{x}_n$] and $\boldsymbol{Y}$=[$\boldsymbol{y}_1$,…, $\boldsymbol{y}_n$] such that the correlation between the canonical component pairs $\left(\boldsymbol{w}_x^T\boldsymbol{X}, \boldsymbol{w}_y^T\boldsymbol{Y}\right)$ are maximized. Initially proposed as a multivariate analysis method by Hotelling [2], recently CCA and its variants have been widely applied to image processing [3,4], image analysis [5,6,7], image retrieval [8], pattern recognition [9,10,11] computer vision [12,13], text analysis and retrieval [14,15], regression and prediction [16,17], information fusion [18], bioinformatics [19] and other fields. However, CCA is, in nature, a linear dimensionality reduction technique, as a result, it can only reveal the linear correlation relationship between two sets of features in a global way, and such a linear model is insufficient to evaluate the nonlinear correlation relationship between features. Let us see an example. Fig.1a and b demonstrate such a phenomenon on the artificial dataset [20]. From Fig.1a and b, we can observe that 1) the 1st pair of canonical components obtained by CCA exhibit nonlinear relationship to some extent (Fig.1a); in other words, the relationship between the 1st pair of canonical components can be approximately deemed to be globally linear; and 2) the linear relationship between the 2nd pair is not significant (Fig.1b). These observations indicate that in the data possibly more complex, nonlinear relationship exists that can not be well discovered by CCA. To attack such nonlinear cases, there are generally three kinds of main approaches proposed so far, i.e. kernel based methods [12,15,21], neural networks [22,23] and locality based methods [24-31,36].

The kernel based CCA [12, 15] and the neural networks based CCA [22,23], as its nonlinear extensions to CCA, are able to deal with the nonlinear correlation problems to some extent. In kernel CCA (KCCA) [12,15], the data is mapped to higher (even infinite) dimensional space (referred as *feature space*) via implicit nonlinear mappings, $\Phi : \boldsymbol{x} \mapsto \Phi\left(\boldsymbol{x}\right)$ and $\Psi : \boldsymbol{y} \mapsto \Psi\left(\boldsymbol{y}\right)$, and a traditional CCA is performed in the feature spaces with the help of so-called “kernel trick” [12,21,45], so a nonlinear problem in the original space is transformed into another more possibly linear one in the feature space so as to discover the

nonlinear correlation hidden between the original data sets. Given an implicit nonlinear mapping $\phi : \boldsymbol{x} \mapsto \phi(\boldsymbol{x})$, the kernel trick can be applied whenever the inner product of the transformed input data, $\phi(\boldsymbol{x})^T \phi(\boldsymbol{y})$, appears in the feature space, where the inner product can be represented in terms of kernel functions in input space, i.e., $\phi(\boldsymbol{x})^T \phi(\boldsymbol{y}) = k(\boldsymbol{x}, \boldsymbol{y})$. A sufficient condition for a kernel function $k(\cdot,\cdot)$ corresponding to an inner product in feature space is given by Mercer's theorem (see, e.g., Ref.[21,45]) [12]. Prominent examples of Mercer kernels are RBF kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\|\boldsymbol{x} - \boldsymbol{y}\|^2 / 2\sigma^2\right)$ and monomial kernel $k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x}^T \boldsymbol{y})^m$, where both σ and m are user specified parameters. Applying kernel trick, we can compute CCA on the mapped data without actually knowing the mappings Φ, Ψ themselves [12]. Let us experimentally evaluate the capability of kernel CCA in reveal the nonlinear correlation between features. For the same artificial dataset [20] mentioned above, the first two canonical component pairs obtained by KCCA using RBF kernels for both $\boldsymbol{X}$ and $\boldsymbol{Y}$ sets are illustrated in Fig.1c and d, respectively, where these canonical component pairs exhibit a more obvious linear relationship in the feature space. The correlation coefficients between these pairs are 0.9627 and 0.8890, respectively, which are higher than the respective values 0.8782 and 0.6328 of CCA in the input space. All these observations indicate that between such input data sets, a nonlinear relationship indeed resides and is better discovered and revealed by KCCA with the help of the nonlinear mappings induced by the RBF kernel. However, the nonlinearity of KCCA is still in a global sense because all the data pairs are transformed by the common mappings, i.e. $\Phi(\cdot)$ and $\Psi(\cdot)$ for $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively, so the linear correlation does not necessarily holds anywhere in the feature space. In other words, the common nonlinear mapping(s) can not necessarily guarantee to transform a nonlinear problem into another linear one in the feature space. Though we can choose a stronger kernel to ensure that the significant linear correlation almost holds between the canonical component pairs in the feature space, however, too strong kernel will inevitably result in undermining the generalization ability [10] for the unseen data. Although CCA and KCCA can also discover the low dimensional manifold to some extent

where high dimensional data lies [12]; however, neither of them takes the local structure of data into account, resulting in the challenge in the case of tackling some complex, nonlinear manifold problems. Further, the choice of both the problem-dependent kernel function and its parameter(s) is still a sticky problem and also a hot research topic [32,33]. On the other hand, the application of neural networks to CCA [22,23] motivates another important advance in nonlinear CCA. The nonlinear correlation can be discovered with the help of the nonlinear processing ability of the neural networks. Unfortunately, the neural networks suffer from some intrinsic problems such as long-time training, slow convergence and local minima. In summary, though both the kernel based methods and neural networks can solve some nonlinear problems, in contrast, CCA integrating locality will more likely be suitable for discovering the local structure hidden in the data.

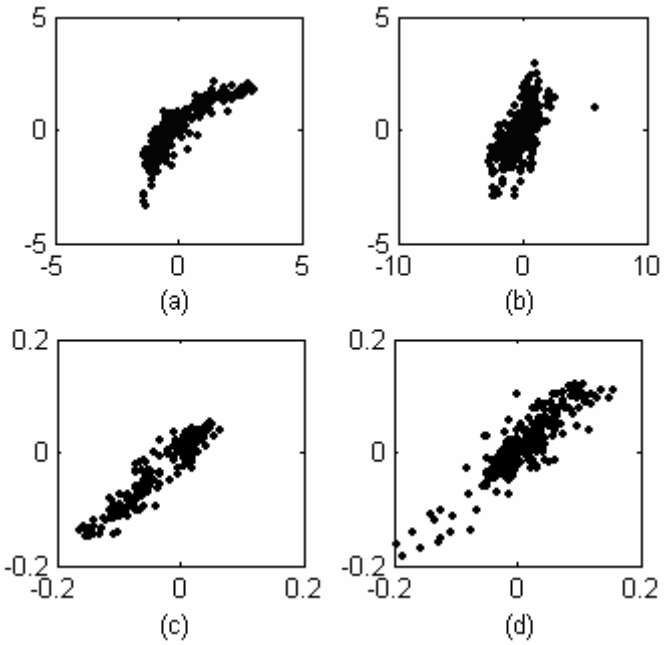


Fig. 1. Sample plots of the 1st and 2nd canonical component pairs obtained by CCA (a-b) [20] and KCCA (c-d), where (a) and (c) denote the 1st pair, (b) and (d) the 2nd pair, respectively. The correlation coefficients between the canonical pairs in (a-d) are 0.8782, 0.6328, 0.9627 and 0.8890, respectively.

Locality based or locality-preserving method is another effective approach to dealing with the nonlinear problem mentioned above. During recent years, remarkable results have been achieved in nonlinear dimensionality reduction by such locality based methods. Tenenbaum et al. proposed Isomap algorithm [24], in which the global geodesic distance between two

points in original space is approximated by the length of the shortest path in the graph of the local neighborhood relationship in low dimensional space. Roweis and Saul proposed locally linear embedding (LLE) [25,36], which assume that any one datum can be reconstructed by its local neighbors in original space and this local reconstruction relationship still holds in low dimensional space. These two approaches are both nonlinear in nature. Recently locality preserving projection (LPP) [26], which aims at keeping the neighborhood relationship in linear dimensionality reduction process, is proposed and applied to data visualization and pattern recognition. The Laplacianface method [27] based on LPP was further developed for face recognition and significant result was achieved. Similarly, locality pursuit embedding (LPE) [28] is developed to preserve as much locality variation information as possible. In brief, for LPP and LPE, the results of the nonlinear dimensionality reduction are obtained by the linear approaches. So both LPP and LPE can be considered as the variation version of PCA. In summary, Isomap, LLE, LPP and LPE share such a characteristic that they preserve the local structure information in original data and thus can discover the low dimensional manifold structure embedded in the original high dimensional space. These locality based approaches are significant development for dimensionality reduction in recent years.

The successful applications of locality based methods to dimensionality reduction inspire us to pay more attention to the locality based methods. Generally, the global nonlinear structures are locally linear and the local structures can be aligned, and many locality based methods embody this heuristic idea [24-31]. In this paper, we incorporate the local structure information into CCA and decompose the globally nonlinear problem into many locally linear ones, consequently, in each small neighborhood field the problem can be treated by linear CCA and the global problem can be solved by optimizing the combination or integration of these local sub-problems. The proposed method, named as locality preserving CCA (LPCCA, for short) takes on the following characters: 1) LPCCA is a locally linear dimensionality reduction approach, yet can yield the effect of globally nonlinear dimensionality reduction. By this approach, the local structure information is preserved and the canonical correlation is also obtained between two data sets; and 2) dimensionality reduction by LPCCA is suitable not only for training samples but also for testing samples. The experiments of data visualization and pose estimation verify its feasibility and

effectiveness. The experiment on COIL-20 dataset shows that pose estimation using LPCCA can obtain higher accuracy than those using CCA and KCCA.

The rest of this paper is organized as follows. In Section 2 both CCA and KCCA are reviewed for comparison. Section 3 is the derivation of LPCCA. The computational issue of LPCCA is discussed in Section 4. The experiments and results are given in Section 5. In Section 6 the conclusions and the future work are discussed.

2. Review of CCA and KCCA

2.1 CCA

Given n pairs of data[*], $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $\boldsymbol{x}_i \in \mathrm{R}^p$, $\boldsymbol{y}_i \in \mathrm{R}^q$, $i=1,\ldots n$, and their mean $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$, CCA aims to find two basis or projection vectors, $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$, respectively for two data sets that maximize the correlation between the random variable $x = \boldsymbol{w}_x^T (\boldsymbol{x}_i - \bar{\boldsymbol{x}})$ and $y = \boldsymbol{w}_y^T (\boldsymbol{y}_i - \bar{\boldsymbol{y}})$, $i=1,\ldots n$, the basis vector pair $(\boldsymbol{w}_x, \boldsymbol{w}_y)$ can be formulated as [1]

$$\begin{aligned}(\boldsymbol{w}_x, \boldsymbol{w}_y) &= \arg\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} \\ &= \arg\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \frac{\sum_{i=1}^n \boldsymbol{w}_x^T (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T \boldsymbol{w}_y}{\sqrt{\sum_{i=1}^n \boldsymbol{w}_x^T (\boldsymbol{x}_i - \bar{\boldsymbol{x}})(\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T \boldsymbol{w}_x} \cdot \sqrt{\sum_{i=1}^n \boldsymbol{w}_y^T (\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T \boldsymbol{w}_y}} \\ &= \arg\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \frac{\boldsymbol{w}_x^T \boldsymbol{X} \boldsymbol{P}_c \boldsymbol{Y}^T \boldsymbol{w}_y}{\sqrt{\boldsymbol{w}_x^T \boldsymbol{X} \boldsymbol{P}_c \boldsymbol{X}^T \boldsymbol{w}_x} \cdot \sqrt{\boldsymbol{w}_y^T \boldsymbol{Y} \boldsymbol{P}_c \boldsymbol{Y}^T \boldsymbol{w}_y}}\end{aligned} \quad (1)$$

Equivalently, $(\boldsymbol{w}_x, \boldsymbol{w}_y)$ can be obtained through solving the following optimization problem with the constraints:

$$\begin{aligned}&\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \boldsymbol{w}_x^T \boldsymbol{X} \boldsymbol{P}_c \boldsymbol{Y}^T \boldsymbol{w}_y \\ &\text{s.t.} \quad \boldsymbol{w}_x^T \boldsymbol{X} \boldsymbol{P}_c \boldsymbol{X}^T \boldsymbol{w}_x = 1, \boldsymbol{w}_y^T \boldsymbol{Y} \boldsymbol{P}_c \boldsymbol{Y}^T \boldsymbol{w}_y = 1\end{aligned} \quad (2)$$

where $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, $\boldsymbol{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$, $\boldsymbol{P}_c = \boldsymbol{I} - \frac{1}{n} \boldsymbol{1}_n \boldsymbol{1}_n^T$, $\boldsymbol{1}_n = [1, \ldots, 1]^T \in R^n$, $\boldsymbol{P}_c$ is a mean-normalization matrix and satisfies $\boldsymbol{P}_c^T = \boldsymbol{P}_c$, $\boldsymbol{P}_c^T \boldsymbol{P}_c = \boldsymbol{P}_c$. Note that in (1), due to the scale invariance of $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$, the constraint condition in (2) is usually (but not necessarily) set as it is [1]. Solving this optimization problem (2), we can obtain the following generalized eigenproblem:

---

[*] In this paper, the scalar is generally written in lowercase and italic font, the vector in lowercase, italic and bold font, and the matrix in capital, italic and bold font.

$$\begin{pmatrix} & \boldsymbol{XP}_c\boldsymbol{Y}^T \\ \boldsymbol{YP}_c\boldsymbol{X}^T & \end{pmatrix}\begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{XP}_c\boldsymbol{X}^T & \\ & \boldsymbol{YP}_c\boldsymbol{Y}^T \end{pmatrix}\begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix} \tag{3}$$

where the eigenvalue λ is just the canonical correlation, i.e., the objective value to be optimized in (2). Eq.(3) can be further decoupled into two generalized eigenproblems w.r.t. $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$, respectively [1]. Once the basis vector pairs, ($\boldsymbol{w}_{xi}$, $\boldsymbol{w}_{yi}$), i=1,…,d, are obtained, the dimensionality reduction of the original data can be performed in the form of $\boldsymbol{W}_x^T\boldsymbol{X}^c$ and $\boldsymbol{W}_y^T\boldsymbol{Y}^c$ to obtain d pairs of canonical components $\left(\boldsymbol{w}_{xi}^T\boldsymbol{X}^c, \boldsymbol{w}_{yi}^T\boldsymbol{Y}^c\right)$, i=1,…,d, for the subsequent computation, where $\boldsymbol{X}^c = \boldsymbol{XP}_c$, $\boldsymbol{Y}^c = \boldsymbol{YP}_c$, denote mean-normalized data matrices, and $\boldsymbol{W}_x = [\boldsymbol{w}_{x1},...,\boldsymbol{w}_{xd}]$, $\boldsymbol{W}_y = [\boldsymbol{w}_{y1},...,\boldsymbol{w}_{yd}]$, separately denoting two projective matrices whose columns correspond to the first d largest common eigenvalues of (3), and d satisfies $d \le \min(p,q)$.

It can be proven that the eigenvector solution of (2), $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$, can be expressed as the linear combination of the centered samples [12], i.e.

$$\boldsymbol{w}_x = \sum_{i=1}^{n}\alpha_i\left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right) = \boldsymbol{X}^c\boldsymbol{\alpha} \quad \text{and} \quad \boldsymbol{w}_y = \sum_{i=1}^{n}\beta_i\left(\boldsymbol{y}_i - \bar{\boldsymbol{y}}\right) = \boldsymbol{Y}^c\boldsymbol{\beta} \tag{4}$$

where $\boldsymbol{\alpha} = [\alpha_1,...,\alpha_n]^T$ and $\boldsymbol{\beta} = [\beta_1,...,\beta_n]^T$ with their respective entries α_i, β_i as the linear combination coefficients. This is usually called “dual representation” [12].

2.2 Kernel CCA (KCCA)

In this case, suppose there are two (possibly) nonlinear mappings: $\Phi : \boldsymbol{x} \mapsto \Phi(\boldsymbol{x})$ and $\Psi : \boldsymbol{y} \mapsto \Psi(\boldsymbol{y})$ to corresponding feature spaces $\mathbb{F}_x$ and $\mathbb{F}_y$, in which the mapped data set now is $\left\{\left(\Phi(\boldsymbol{x}_i), \Psi(\boldsymbol{y}_i)\right)\right\}_{i=1}^{n}$, kernel CCA (KCCA) is actually traditional CCA using the mapped data. Similar to CCA, the basis vector pair $\left(\boldsymbol{w}_{\phi,x}, \boldsymbol{w}_{\psi,y}\right)$ in the kernel feature space has the corresponding dual representations below:

$$\boldsymbol{w}_{\phi,x} = \sum_{i=1}^{n}\alpha_i\Phi(\boldsymbol{x}_i) = \Phi(\boldsymbol{X})\boldsymbol{\alpha}, \text{ and } \boldsymbol{w}_{\psi,y} = \sum_{i=1}^{n}\beta_i\Psi(\boldsymbol{y}_i) = \Psi(\boldsymbol{Y})\boldsymbol{\beta} \tag{5}$$

where $\Phi(\boldsymbol{X}) = \left[\Phi(\boldsymbol{x}_1),...,\Phi(\boldsymbol{x}_n)\right]$, $\Psi(\boldsymbol{Y}) = \left[\Psi(\boldsymbol{y}_1),...,\Psi(\boldsymbol{y}_n)\right]$ and the means

$\bar{\Phi}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n}\Phi(\boldsymbol{x}_i)$ and $\bar{\Psi}(\boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n}\Psi(\boldsymbol{y}_i)$ in kernel feature space are now tentatively assumed to be zero for derivational convenience (in fact the centering process of samples in feature space can be performed in a similar way to the proposed method in [45]). In addition, here we abuse the notation $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_n]^T$ and $\boldsymbol{\beta} = [\beta_1, ..., \beta_n]^T$ to still denote corresponding dual coefficient vectors in feature spaces.

Just like the form of the objective function described in (2) for CCA, the objective function to be maximized for KCCA can be written as $\boldsymbol{w}_{\phi,x}^T \Phi(\boldsymbol{X}) \Psi(\boldsymbol{Y})^T \boldsymbol{w}_{\psi,y}$. Inserting the dual representations in (5) into it, the objective function can be rewritten as $\boldsymbol{\alpha}^T \Phi(\boldsymbol{X})^T \Phi(\boldsymbol{X}) \Psi(\boldsymbol{Y})^T \Psi(\boldsymbol{Y}) \boldsymbol{\beta}$, where the inner products appear in the expression and thus the kernel trick [12,21,45] can be employed as described in the introduction. Let us define the kernel matrices $\boldsymbol{K}_x$, $\boldsymbol{K}_y$ $\in R^{n \times n}$ with the *ij*-th entry denoted respectively by $(\boldsymbol{K}_x)_{ij} = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$ and $(\boldsymbol{K}_y)_{ij} = \Psi(\boldsymbol{y}_i)^T \Psi(\boldsymbol{y}_j)$, then the objective function, $\boldsymbol{\alpha}^T \Phi(\boldsymbol{X})^T \Phi(\boldsymbol{X}) \Psi(\boldsymbol{Y})^T \Psi(\boldsymbol{Y}) \boldsymbol{\beta}$, of KCCA is now simplified to $\boldsymbol{\alpha}^T \boldsymbol{K}_x \boldsymbol{K}_y \boldsymbol{\beta}$. Similarly, applying the same trick to the constraint condition described in (2), thus KCCA can be reformulated as the following optimization problem:

$$\begin{aligned} &\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \boldsymbol{\alpha}^T \boldsymbol{K}_x \boldsymbol{K}_y \boldsymbol{\beta} \\ &\text{s.t.} \quad \boldsymbol{\alpha}^T \boldsymbol{K}_x \boldsymbol{K}_x \boldsymbol{\alpha} = 1, \boldsymbol{\beta}^T \boldsymbol{K}_y \boldsymbol{K}_y \boldsymbol{\beta} = 1 \end{aligned} \tag{6}$$

Using the same optimization strategy as CCA [1], we can obtain the following generalized eigenproblem of KCCA:

$$\begin{pmatrix} & \boldsymbol{K}_x \boldsymbol{K}_y \\ \boldsymbol{K}_y \boldsymbol{K}_x & \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{K}_x^2 & \\ & \boldsymbol{K}_y^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{pmatrix} \tag{7}$$

In general, the regularization techniques [8,10,12,15,34,35] is employed in solving (7) to avoid the singularity problem. Once the dual solution vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ have been obtained, for any one sample $\boldsymbol{x}$, its projection in the feature space $\mathbb{F}_x$, which is induced by kernel K_x, can be formulated as

$$\boldsymbol{w}_{\phi,x}^T \Phi(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K_x(\boldsymbol{x}_i, \boldsymbol{x}) \tag{8}$$

where the kernel trick is also employed. The projection $\boldsymbol{y}$ in its feature space $\mathbb{F}_y$, which is induced by kernel K_y, can be derived in a similar way.

3. Locality Preserving CCA (LPCCA)

As mentioned in the introduction, kernel based and locality based methods are two effective approaches to dealing with the nonlinear problems. In Section 2.2 we have reviewed KCCA as one of the nonlinear extension of CCA. In this section, the locality is introduced into CCA and locality based method of CCA, namely locality preserving CCA (LPCCA) is proposed as another nonlinear extension of CCA. Firstly a key equivalent description of CCA is given in Section 3.1; then in Section 3.2 we introduce locality into CCA; at last we obtain a formula description of LPCCA in Section 3.3.

3.1 An equivalent description of CCA

The optimization problem of CCA can be written in another equivalent form [37] as:

$$\begin{aligned} &\min_{\boldsymbol{w}_x,\boldsymbol{w}_y} \sum_{i=1}^{n} \left\| \boldsymbol{w}_x^T \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}} \right) - \boldsymbol{w}_y^T \left(\boldsymbol{y}_i - \bar{\boldsymbol{y}} \right) \right\|^2 \\ &\text{s.t.} \quad \sum_{i=1}^{n} \left\| \boldsymbol{w}_x^T \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}} \right) \right\|^2 = 1, \ \sum_{i=1}^{n} \left\| \boldsymbol{w}_y^T \left(\boldsymbol{y}_i - \bar{\boldsymbol{y}} \right) \right\|^2 = 1 \end{aligned} \tag{9}$$

where the objective function can be expanded as

$$\begin{aligned} &\sum_{i=1}^{n} \left\| \boldsymbol{w}_x^T \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}} \right) - \boldsymbol{w}_y^T \left(\boldsymbol{y}_i - \bar{\boldsymbol{y}} \right) \right\|^2 \\ &= \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} \left\{ \boldsymbol{w}_x^T \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)^T \boldsymbol{w}_x + \boldsymbol{w}_y^T \left(\boldsymbol{y}_i - \boldsymbol{y}_j \right) \left(\boldsymbol{y}_i - \boldsymbol{y}_j \right)^T \boldsymbol{w}_y \right\} \\ &\quad - \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{n} 2 \boldsymbol{w}_x^T \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \left(\boldsymbol{y}_i - \boldsymbol{y}_j \right)^T \boldsymbol{w}_y \end{aligned} \tag{10}$$

It can easily be proven that (9) can be expressed as the following equivalent form:

$$\begin{aligned} &\max_{\boldsymbol{w}_x,\boldsymbol{w}_y} \boldsymbol{w}_x^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \left(\boldsymbol{y}_i - \boldsymbol{y}_j \right)^T \cdot \boldsymbol{w}_y \\ &\text{s.t.} \quad \boldsymbol{w}_x^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)^T \cdot \boldsymbol{w}_x = 1 \\ &\qquad \boldsymbol{w}_y^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\boldsymbol{y}_i - \boldsymbol{y}_j \right) \left(\boldsymbol{y}_i - \boldsymbol{y}_j \right)^T \cdot \boldsymbol{w}_y = 1 \end{aligned} \tag{11}$$

where the constant 1/2n has been ignored.

3.2 Introduction of locality

Note that in (11), the canonical correlation is defined in terms of the projections of the differences between all the sample pairs. This definition holds for the simple linear canonical correlation, however, for the complex, nonlinear correlation cases mentioned in the introduction, i.e., the nonlinear low dimensional manifold embedded in the ambient space, the linear canonical correlation makes sense only in a local field. For any given one sample pair ($\boldsymbol{x}_i$, $\boldsymbol{y}_i$), the locally linear correlation can be expressed as $\boldsymbol{w}_x^T \cdot \sum_{j\in \mathrm{ne}(i)} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)^T \cdot \boldsymbol{w}_y$, where ne($i$) denotes the index set for the local neighbors of $\boldsymbol{x}_i$ (or $\boldsymbol{y}_i$). The local neighbor of $\boldsymbol{x}_i$ can be defined in the following two ways [26,27,38]:

a). ε-hypersphere definition: if $\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\| \le \varepsilon$, then we say $\boldsymbol{x}_j$ is the local neighbor of $\boldsymbol{x}_i$, where ε is a user specified threshold.

b). k-nearest neighborhood definition: if $\boldsymbol{x}_j$ is among the k-nearest neighbors of $\boldsymbol{x}_i$, we say $\boldsymbol{x}_j$ is the local neighbor of $\boldsymbol{x}_i$.

The definitions a) and b) are also suitable for $\boldsymbol{Y}$ data set. Furthermore, we let LN($\boldsymbol{x}_i$) denotes the sample set that comprises the local neighbors of $\boldsymbol{x}_i$, thus $\boldsymbol{x}_j \in \mathrm{LN}(\boldsymbol{x}_i)$ iff $j \in \mathrm{ne}(i)$. On the basis of definition of local neighborhood, we define the similarity matrices $\boldsymbol{S}_x = \left\{S_{ij}^x\right\}_{i,j=1}^n$ and $\boldsymbol{S}_y = \left\{S_{ij}^y\right\}_{i,j=1}^n$, where

$$S_{ij}^x = \begin{cases} \exp\left(-\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / t_x\right), \text{if } \boldsymbol{x}_j \in \mathrm{LN}\left(\boldsymbol{x}_i\right) \text{ or } \boldsymbol{x}_i \in \mathrm{LN}\left(\boldsymbol{x}_j\right) \\ 0, \text{ otherwise} \end{cases} \tag{12a}$$

$$S_{ij}^y = \begin{cases} \exp\left(-\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2 / t_y\right), \text{if } \boldsymbol{x}_j \in \mathrm{LN}\left(\boldsymbol{x}_i\right) \text{ or } \boldsymbol{x}_i \in \mathrm{LN}\left(\boldsymbol{x}_j\right) \\ 0, \text{ otherwise} \end{cases} \tag{12b}$$

The parameters t_x is generally taken as the mean square distance $\sum_{i=1}^n \sum_{j=1}^n 2\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / n(n-1)$ or the number with the same magnitude, and things are similar for t_y. In fact, (12) reflects the locality around each data point. Obviously, the smaller the $\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|$ ($\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|$), the closer

they, and thus the larger S_{ij}^{x} (S_{ij}^{y}). Again from the definition of (12) we know that $\boldsymbol{S}_x$ and $\boldsymbol{S}_y$ are symmetric, sparse matrices. Using the definitions of local neighborhood and similarity defined above, we can write the canonical correlation in a local field as $\boldsymbol{w}_x^T \cdot \sum_{j=1}^{n} S_{ij}^{x} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) S_{ij}^{y} \left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)^T \cdot \boldsymbol{w}_y$. Now the globally nonlinear problem can be decomposed into *n* locally linear sub-problems, and reversely combining all these sub-problems together can approximate the original problem. So CCA incorporating the local information can be written as:

$$\begin{aligned} &\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \boldsymbol{w}_x^T \cdot \sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}^{x} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) S_{ij}^{y} \left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)^T \cdot \boldsymbol{w}_y \\ &\text{s.t.} \quad \boldsymbol{w}_x^T \cdot \sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}^{x2} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)\left(\boldsymbol{x}_i - \boldsymbol{x}_j\right)^T \cdot \boldsymbol{w}_x = 1 \\ &\qquad \boldsymbol{w}_y^T \cdot \sum_{i=1}^{n}\sum_{j=1}^{n} S_{ij}^{y2} \left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)^T \cdot \boldsymbol{w}_y = 1 \end{aligned} \tag{13}$$

3.3 Derivation of LPCCA

The optimization problem (13) can be rewritten after some algebraic manipulations as (ignoring the trivial constants)

$$\begin{aligned} &\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \boldsymbol{w}_x^T \boldsymbol{X} \boldsymbol{S}_{xy} \boldsymbol{Y}^T \boldsymbol{w}_y \\ &\text{s.t.} \quad \boldsymbol{w}_x^T \boldsymbol{X} \boldsymbol{S}_{xx} \boldsymbol{X}^T \boldsymbol{w}_x = 1 \\ &\qquad \boldsymbol{w}_y^T \boldsymbol{Y} \boldsymbol{S}_{yy} \boldsymbol{Y}^T \boldsymbol{w}_y = 1 \end{aligned} \tag{14}$$

where $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n], \boldsymbol{Y}=[\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n]$, $\boldsymbol{S}_{xx} = \boldsymbol{D}_{xx} - \boldsymbol{S}_x \circ \boldsymbol{S}_x$, $\boldsymbol{S}_{yy} = \boldsymbol{D}_{yy} - \boldsymbol{S}_y \circ \boldsymbol{S}_y$, $\boldsymbol{S}_{xy} = \boldsymbol{D}_{xy} - \boldsymbol{S}_x \circ \boldsymbol{S}_y$, the symbol $\circ$ denotes an operator, for matrices $\boldsymbol{A}$, $\boldsymbol{B}$ with the same size, $\left(\boldsymbol{A} \circ \boldsymbol{B}\right)_{ij} = \boldsymbol{A}_{ij}\boldsymbol{B}_{ij}$, and $\boldsymbol{A}_{ij}$ denotes the *ij*-th entry of $\boldsymbol{A}$, $\boldsymbol{D}_{xx}$ ($\boldsymbol{D}_{yy}$, $\boldsymbol{D}_{xy}$) is a diagonal matrix of size *n*-by-*n*, and its *i*th diagonal entry equals the sum of the entries in the *i*th row (or the *i*th column due to the symmetry) of the matrix $\boldsymbol{S}_x \circ \boldsymbol{S}_x$ ($\boldsymbol{S}_y \circ \boldsymbol{S}_y$, $\boldsymbol{S}_x \circ \boldsymbol{S}_y$). For example, let $\boldsymbol{S}=\boldsymbol{S}_x \circ \boldsymbol{S}_x$, $\boldsymbol{S} \in R^{n \times n}$, then $\boldsymbol{D}_{xx}=\text{diag}(\sum_{j=1}^{n} S_{1j},\ldots,\sum_{i=1}^{n} S_{nj})=\text{diag}(\sum_{i=1}^{n} S_{i1},\ldots,\sum_{i=1}^{n} S_{in})$, and the similar expressions exist for $\boldsymbol{D}_{yy}$ and $\boldsymbol{D}_{xy}$. The definition of $\boldsymbol{S}_{xx}$ ($\boldsymbol{S}_{yy}$, $\boldsymbol{S}_{xy}$) is similar to that of Laplace matrix in LPP [26] except that in LPCCA the Laplace matrix is based on the computational result of operator $\circ$

on $\boldsymbol{S}_x$ and/or $\boldsymbol{S}_y$. $\boldsymbol{S}_{xx}$, $\boldsymbol{S}_{yy}$ and $\boldsymbol{S}_{xy}$ are all symmetric, and $\boldsymbol{S}_{xy}$= $\boldsymbol{S}_{yx}$ if we define $\boldsymbol{S}_{yx} = \boldsymbol{D}_{yx} - \boldsymbol{S}_y \circ \boldsymbol{S}_x$, where $\boldsymbol{D}_{yx}$ is diagonal, and its ith diagonal entry equals the sum of the entries in the *i*th row (or the *i*th column) of $\boldsymbol{S}_y \circ \boldsymbol{S}_x$.

Solving the optimization problem (14) by utilizing the same optimization strategy as that of CCA [1], we obtain

$$\begin{pmatrix} & \boldsymbol{X}\boldsymbol{S}_{xy}\boldsymbol{Y}^T \\ \boldsymbol{Y}\boldsymbol{S}_{yx}\boldsymbol{X}^T & \end{pmatrix}\begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{X}\boldsymbol{S}_{xx}\boldsymbol{X}^T & \\ & \boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T \end{pmatrix}\begin{pmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{pmatrix} \tag{15}$$

Once the basis vector pairs ($\boldsymbol{w}_x$, $\boldsymbol{w}_y$) is obtained, the dimensionality reduction can be performed in the form of $\boldsymbol{w}_x^T\boldsymbol{x}$ and $\boldsymbol{w}_y^T\boldsymbol{y}$. During the computation, we keep the local information S_{ij}^x and S_{ij}^y unchanged, and we attempt to ensure that if the data points are close in the original input space, then after dimensionality reduction using LPCCA, the data points in the projection space with reduced dimensionality are still close. This will be validated by the data visualization experiment (see Section 5.1 later). The primary equation of LPCCA, (15), is similar to those of CCA and KCCA, i.e., (3) and (7). However, the data need not be mean-normalized as in CCA and KCCA.

If all the entries of $\boldsymbol{S}_x$ and $\boldsymbol{S}_y$ are set to 1 (or 1/*n*), $\boldsymbol{S}_{xx}$, $\boldsymbol{S}_{yy}$, $\boldsymbol{S}_{xy}$ and $\boldsymbol{S}_{yx}$ all equal to $n\boldsymbol{P}_c$ (or $\boldsymbol{P}_c/n$). Ignoring the trivial coefficient *n* (or 1/*n*), (15) degrades into (3). So LPCCA generalizes CCA, and CCA is the special case of the former.

4. Computational issue of LPCCA

In real applications, we often encounter such cases that a large amount of high dimensional data points correspond some underlying, varying parameter(s). For example, given a series of face images corresponding to pose parameters varying in pan and tilt, the face images, in fact, usually lie in a low dimensional nonlinear manifold embedded in the ambient space [12,24-26,39]. Without loss of generality, the *n* face images (by concatenating each column of an image to form a *p* dimensional vector) is taken as $\boldsymbol{X}$ set, and the corresponding *n* vectors of pose parameters (or else parameters) of size *q* are taken as $\boldsymbol{Y}$ set, in general $p>>n$ and $q<<n$. For this typical application, a direct computation using (15) is difficult (even infeasible) due

to two large matrices of size $(p+q)\times(p+q)$ involved in the generalized eigenproblem (15). To overcome this computational issue, we proposed the following feasible algorithm. Firstly we give the following dual theorem:

Theorem 1 (dual theorem): Given original sample set $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n] \in R^{p\times n}$ and $\boldsymbol{Y}=[\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n]\in R^{q\times n}$, the solution of LPCCA, as described in (15), $\boldsymbol{w}_x$ and $\boldsymbol{w}_y$, must lie in the subspaces respectively spanned by $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ and $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_n$, in other words, $\boldsymbol{w}_x=\boldsymbol{X}\boldsymbol{\alpha}$ and $\boldsymbol{w}_y=\boldsymbol{Y}\boldsymbol{\beta}$ hold simultaneously, where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ both denote corresponding combination coefficient vectors of size n (The proof is given in Appendix. Note that both $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ do not equal to their counterparts in (4) or (5), yet we still abuse them for convenience.)

According to the dual theorem, the optimization problem of LPCCA (15) can be reformulated as follows:

$$\begin{aligned}&\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{\beta}\\ &\text{s.t.}\quad \boldsymbol{\alpha}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S}_{xx}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}=1,\ \boldsymbol{\beta}^T\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{\beta}=1\end{aligned} \tag{16}$$

Solving this optimization problem using the same optimization strategy as CCA [1], we obtain the following generalized eigenproblem w.r.t. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$:

$$\begin{cases}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{\beta}=\lambda\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S}_{xx}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha} & \text{(17a)}\\ \boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yx}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\alpha}=\lambda\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{\beta} & \text{(17b)}\end{cases}$$

Merging these two equations together, we can obtain the following generalized eigenproblem w.r.t. $\boldsymbol{\beta}$:

$$\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{\beta}=\lambda^2\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{\beta} \tag{18}$$

where $\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{Y}$ and $\boldsymbol{Y}^T\boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{Y}$ are matrices of size n-by-n. Thus when n (>>q) is very large, solving the eigen-system is time-consuming. To reduce the computational burden, we adopt the following technique: left multipling both sides of (18) by $\left(\boldsymbol{Y}\boldsymbol{Y}^T\right)^{-1}\boldsymbol{Y}$ (note that $\boldsymbol{Y}\boldsymbol{Y}^T$ is a matrix of size $q\times q$ and generally invertible due to q<<n) and rewriting $\boldsymbol{Y}\boldsymbol{\beta}$ as $\boldsymbol{w}_y$, we can obtain

$$\boldsymbol{Y}\boldsymbol{S}_{yx}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{w}_y=\lambda^2\boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{w}_y \tag{19}$$

This is an equivalent but scale-reduced generalized eigenproblem, in which the size of two

associated matrices is only $q \times q$, as a result, it is computationally easy to obtain its eigen-pairs $\left(\lambda_i, \boldsymbol{w}_{yi}\right)$, i=1,…,q. From (17a), we have

$$\boldsymbol{\alpha} = \frac{1}{\lambda}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{w}_y \qquad (20)$$

where $\boldsymbol{X}^T\boldsymbol{X} \in R^{n\times n}$ is generally invertible due to $n<<p$. Finally, we obtain the ith basis vector w.r.t. $\boldsymbol{X}$ set, $\boldsymbol{w}_{xi}$, as follows

$$\boldsymbol{w}_{xi} = \boldsymbol{X}\boldsymbol{\alpha} = \frac{1}{\lambda_i}\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{S}_{xx}^{-1}\boldsymbol{S}_{xy}\boldsymbol{Y}^T\boldsymbol{w}_{yi},\ i=1,\ldots,q \qquad (21)$$

Now we obtain the basis vector $\boldsymbol{w}_x$. It is worth noting that when $\boldsymbol{S}_{xx}$ is singular, we generally substitute $\boldsymbol{S}_{xx}$ with $\boldsymbol{S}_{xx}+\mu\boldsymbol{I}$ to avoid its singularity as well as ensure the computational stability [34], where $\boldsymbol{I}$ is the identity matrix and μ is a small non-negative number and taken as 0.001 in the following experiments.

5. Experiments and analysis

In this section, we design two experiments using LPCCA, namely data visualization and pose estimation. Firstly we validate the preservation of local structure information of data using data visualization experiment, and then we perform pose estimation experiment to evaluate the ability of LPCCA to prediction by comparing with that of CCA and KCCA.

5.1 Data visualization

In this experiment, we employ an artificial face image data which contains 698 face images of size 64×64. The pose parameters of face images vary in pan ranging approximately within ±75 degree, tilt ranging approximately within ±10 degree and different illumination. This data is once used in [24] to discover the *intrinsic degree of freedom* of dataset. In this experiment, face images are taken as $\boldsymbol{X}$ set, pose parameters and illumination as $\boldsymbol{Y}$ set. The local neighbor is defined according to k-nearest neighborhood definition (i.e., the definition b) in Section 3.2), where k is set to 5. After a group of basis vectors $\boldsymbol{W}_x$=[$\boldsymbol{w}_{x1}$, $\boldsymbol{w}_{x2}$, $\boldsymbol{w}_{x3}$] are obtained, the face images are projected onto $\boldsymbol{W}_x$ to generate 698 three-dimensional feature vectors. The data points with the first 2 features obtained by the projections on $\boldsymbol{w}_{x1}$ and $\boldsymbol{w}_{x2}$, which corresponds to the first two largest canonical correlations, as well as their corresponding face

images, are illustrated in Fig. 2.

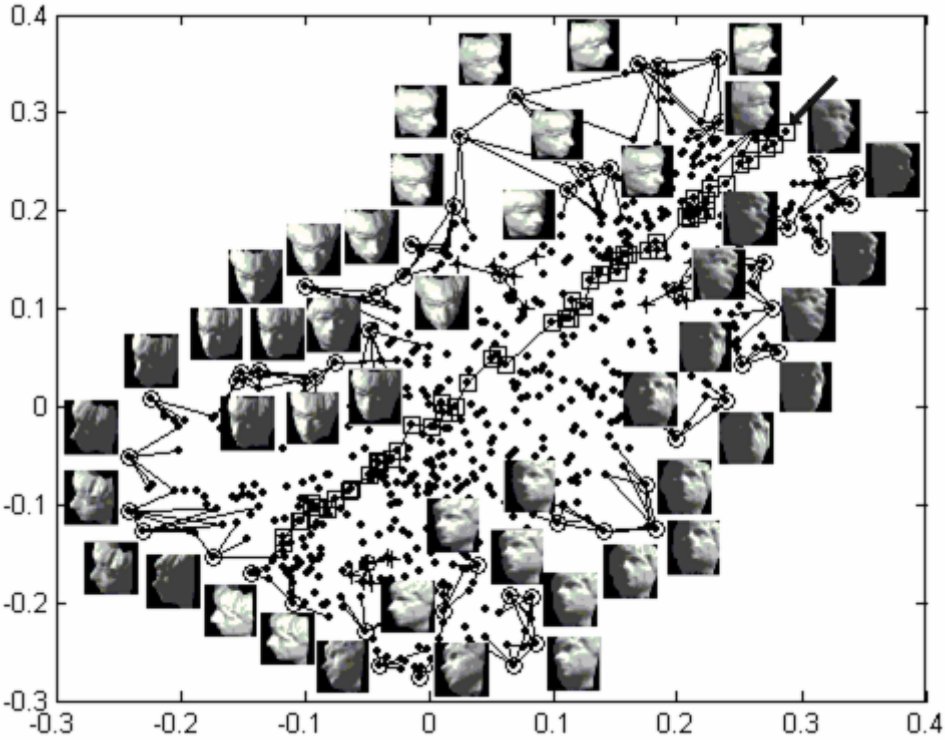


Fig. 2. The two dimensional data visualization after performing LPCCA on face image dataset. The horizontal and vertical coordinate denote the first and second feature value, respectively. The black dots denote the data points after dimensionality reduction. The face beside the circle denotes the corresponding face image associated with this data point. All these data points associated with face images are connected with their 5-nearest neighbors to build up a local neighborhood graph. In the midst of the figure, the face images corresponding to 3 circles along with their respective 5-nearest neighbors (denoted by symbol +) are given in Fig.3. Moreover, the variation of the pan pose parameters corresponding to squared data points (along the direction of arrow in figure) is given in Fig. 6.

From Fig. 2 we can obtain the following two observations: 1) in a relative large range (e.g. along the up-right toward bottom-left of the Fig. 2), an obvious canonical correlation is seen between the features and their pose parameters; in other words, the pose parameters of the corresponding face images vary successively along the arrow direction; meanwhile, the tilt pose parameters vary successively along the direction of up-left toward bottom-right; and 2) if the data points are the neighbors around some data point in reduced dimension space (see Fig. 2, the circled data points as well as their respective 5-nearest neighbors), then these corresponding face images (as seen in Fig. 3) seem also to keep the neighbor relationship, i.e. they are highly similar to each other in pose. This fact shows that LPCCA still preserves the

local structure characteristic after the dimensionality reduction. From the observations 1) and 2) we can conclude that dimensionality reduction using LPCCA can not only capture the global variation of data (in [24] this global variation is referred as intrinsic dimensionality), but also preserve the local structure information. In Fig. 4 the local neighborhood relationship for all data points in low dimensional space are given. As shown in Fig.4, some data points cluster together in some local fields and represent some local properties, e.g. in the circled field near the up-left corner in Fig. 4, the illumination of the face images is obviously weak (see also Fig. 2).

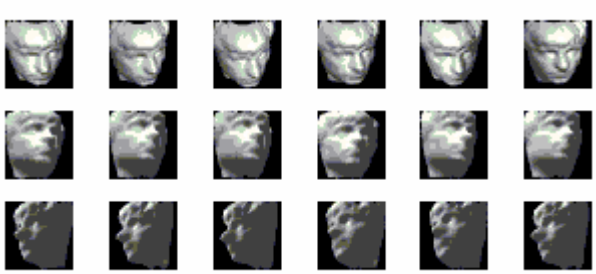


Fig. 3. The face images corresponding to 3 circled data points along with their respective 5-nearest neighbors (symbol +) in Fig. 2. In each row, the left first image denotes the circled data point and the rest 5 images of each row denote its 5-nearest neighbors in projected low dimensional space.

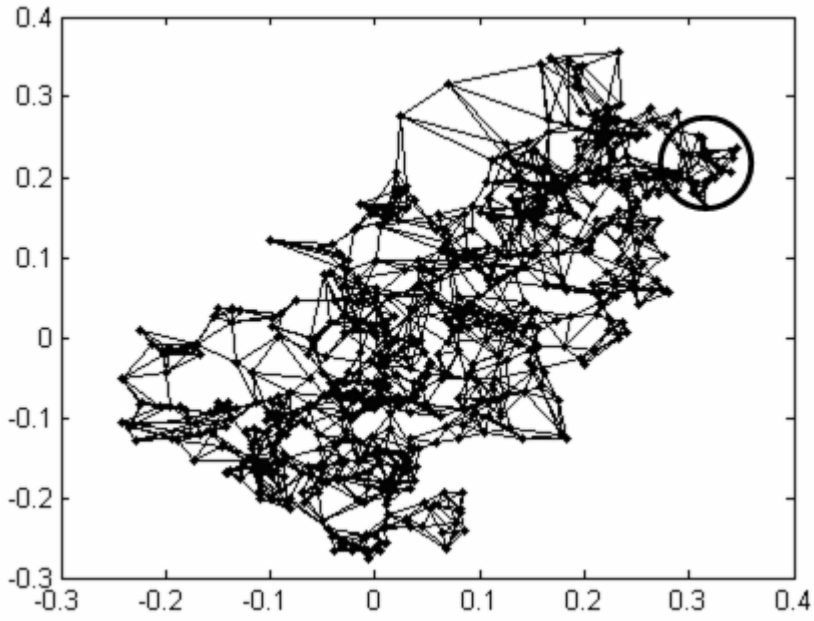


Fig. 4. The two dimensional data visualization after performing LPCCA on face image dataset. The horizontal and vertical axes denote the first and second feature value, respectively. The black dots denote the data points after dimensionality reduction. All the data points are connected with their 5-nearest neighbors by solid lines to build up a local neighboring graph.

In contrast to LPCCA, the dimensionality reduction using CCA is also performed on this face data. The two dimensional data visualization and some face images corresponding to part of data are given in Fig. 5. In Fig. 5 we can obtain the following observations: the distribution of the data points with reduced dimensionality correspond to the successive change of pose parameters of face images. More specifically, approximately along the horizontal direction, the pose parameters vary in pan; while approximately along the vertical direction, the pose parameters vary in tilt. This phenomenon is similar to that in LPCCA and [24]. Moreover, we also investigate the similarity among the neighborhoods in reduced dimensionality space. In Fig. 6 the face images corresponding to some circled points in Fig. 5 along with their respective 5-nearest neighbors are given. Visually, there also exists similarity between the circled point and the respective 5 nearest neighbors.

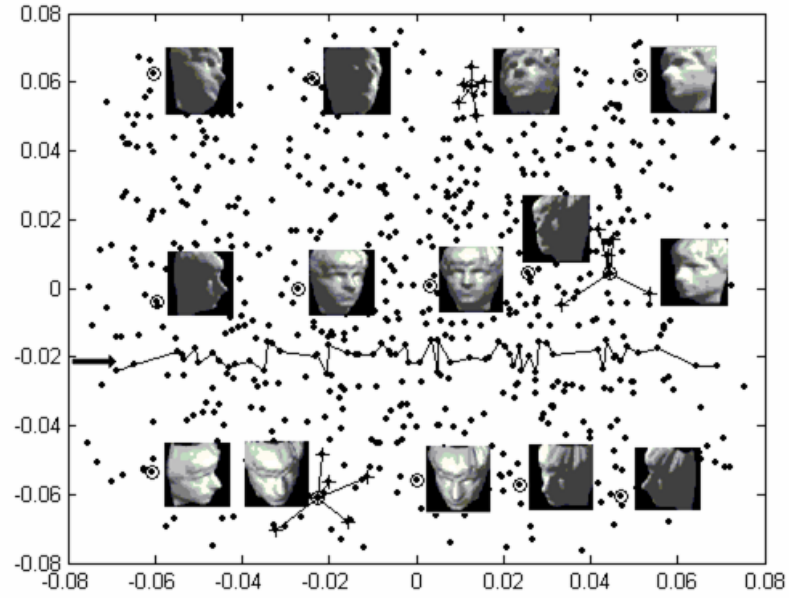


Fig. 5. The two dimensional data visualization after performing CCA on face image dataset. The horizontal and vertical coordinate denote the first and second feature value, respectively. The black dots denote the data points after dimensionality reduction. The face beside the circle denotes the corresponding face image to this data.

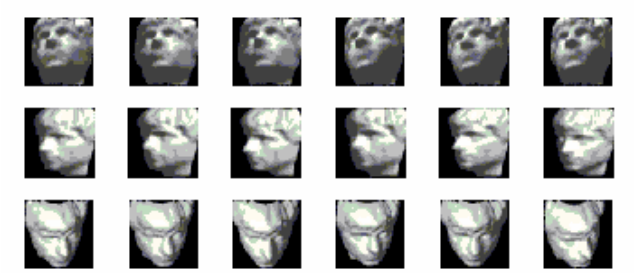


Fig. 6. The face images corresponding to 3 circled data points along with their respective

5-nearest neighbors (symbol +) in Fig. 5. On each row, the left first image denotes the circled data point and the rest 5 images denote its 5-nearest neighbors in low dimensional space.

After analysis of the characteristics of the data distribution and the local similarity in CCA and LPCCA, we now investigate the difference of the pose change incurred before and after the introduction of locality information. From discussion above, it is worth pointing out that the principal directions of pose change for LPCCA is approximately along the diagonals of the figure (see Fig.2), while along the horizontal axis direction of the figure for CCA (see Fig.5). We take representative sections along the arrows in Fig.2 and Fig.5, respectively, and investigate difference between the pose changes in pan for the data near the sections and illustrate this difference in Fig. 7. From Fig.7 we can obtain the following observations: 1) the pose change in pan globally in order both for CCA and LPCCA; and 2) the pose parameter corresponding to CCA locally fluctuates more heavily than LPCCA. In other words, the local structure is better preserved in LPCCA by the introduction of the locality based method, which indicates that LPCCA is possibly more suitable for pose estimation.

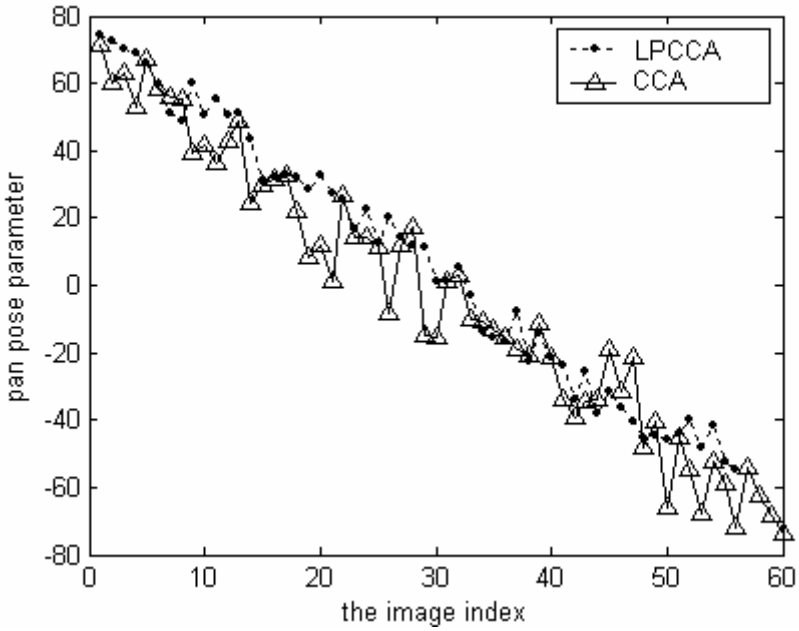


Fig. 7. The variation of the pose parameter in pan for the data points near the section planes in Fig.2 and Fig. 5, respectively. The horizontal coordinate (i.e. the image index) denotes the serial number of data point along the arrows in Fig.2 and Fig. 5, respectively.

5.2 Pose estimation

One of the primary goals of an intelligent vision system is to recognize objects in an image

and compute their poses in the three-dimensional scene. Such a recognition system has wide applications ranging from visual inspection, robot vision to autonomous navigation [40,41]. So pose estimation has become one of the active research topics in computer vision field [12, 40, 42]. Murase and Nayar [40] studied the applications of PCA eigenspace method to pattern recognition and pose estimation and established a framework for pose estimation. Melzer et al. [12] studied the pose estimation using KCCA and PCA and the experiment results show that the estimation accuracy using KCCA outperforms PCA.

Given a serials of images and their corresponding pose parameters, for a new or unseen sample, its unknown pose parameters are estimated by the follow strategy [12,40]:

1). Perform dimensionality reduction on the image samples by some approach (e.g. KCCA) to *d* dimension.

2). By resampling method (e.g. cubic spline interpolation), perform mapping from pose parameter space Ω to projection space $\Re^d$ to construct the parametric manifold [12,40,43].

3). Perform dimensionality reduction to the new sample $\boldsymbol{x}$ and then find its nearest neighborhood $\hat{\boldsymbol{x}}$ in $\Re^d$, the corresponding pose parameter $\hat{\theta}$ is just the estimated pose parameter.

In this experiment, step 2 is modified such that only the neighbors of the new sample rather than the whole samples are selected to construct the parametric manifold, and the reason of doing so is due to the locality-preserving property of LPCCA. Such a modification brings us two advantages: 1) improvement of accuracy for pose estimation, because resampling based on only the neighbors of the new sample can utilize the locally linear structure thus possibly improve the estimation accuracy; and 2) reduction in computation, because only partial samples rather than all ones are involved in the pose computation such that the computational burden can be alleviated.

In this experiment, often-used COIL-20 dataset [44] is employed (free available at http://www1.cs.columbia.edu/CAVE/research/softlib/coil-20.html), which contains total 1440 images of size 128×128 with black background for 20 different subjects. For each subject, the camera moves around it in pan at interval of 5 degree and takes total 72 different images. The

sample images for 20 subjects are shown in Fig. 8a.

(a) (b)

Fig. 8. (a) the sample images for 20 subjects in COIL-20 dataset and (b) the training sample images for subject 1 when parameter L=3.

For each subject, we select every Lth image along pose parameter line for training, i.e. the 1st, Lth, 2Lth,…, 「72/L」×Lth images for training, the rest for testing (the samples to be estimated). We set respectively L to 2 and 3 in two groups of experiments such that the pose parameters of the training samples are degrees of 0, 5, 15, 25, …, 345, 355 and of 0, 10, 25, 40, …, 340, 355, with the "pose resolution" of approximately 10 and 15 degrees, respectively. Note that the first image of each subject (whose pose parameter is set to 0 in degree) is always taken as training sample to act as a "boundary condition" of the interpolation during pose estimation. For any one subject, the numbers of training/testing set are 37/35 and 25/47 when L is taken as 2 and 3, respectively. The training sample images for subject 1 and the case of L=3 are given in Fig. 8b. In computation, we take the training image data as $\boldsymbol{X}$ set and pose parameters as $\boldsymbol{Y}$ set. The pose estimation of any subject is performed independently. For example, for subject 1 and the case of L=3, let $\boldsymbol{X}$=[$\boldsymbol{x}_1$,…, $\boldsymbol{x}_{25}$] and $\boldsymbol{Y}$=[$\boldsymbol{y}_1$,…, $\boldsymbol{y}_{25}$], where $\boldsymbol{x}_i$ denotes a vector concatenated column by column from the ith image data, and $\boldsymbol{y}_i = [\sin\theta_i, \cos\theta_i]^T$ *, and θ_i denotes the pose parameter of $\boldsymbol{x}_i$. For any one subject, in each experiment, the estimation error is defined by $\Delta\vartheta_i = \left|\hat{\vartheta}_i - \vartheta_i\right|, i = 1,...,T$, where $\hat{\vartheta}_i$ and

* In [12], the authors found that directly using the scalar θ_i to represent $\boldsymbol{y}_i$ yields a discontinuity at $\boldsymbol{y}_i$=360 degree. For this reason they choose a periodic, trigonometric representation of pose parameter $\boldsymbol{y}_i = [\sin\theta_i, \cos\theta_i]^T$. Single $\boldsymbol{y}_i$ can determine only one pose parameter θ_i . Here we followed this strategy.

ϑ_i respectively denote the estimated pose parameter of the *i*th image and its true value (in degree), and *T* the testing set size (equals to 35 and 47 for *L*=2 and 3, respectively). Moreover, the mean of estimation errors $\overline{\Delta\vartheta} = \frac{1}{T}\sum_{i=1}^{T}\Delta\vartheta_i$ and its standard deviation $\mathrm{std}(\Delta\vartheta) = \sqrt{\frac{1}{T-1}\sum_{i=1}^{T}\left(\Delta\vartheta_i - \overline{\Delta\vartheta}\right)^2}$ are taken as the criteria to evaluate the performance of the pose estimation.

CCA and KCCA are also performed for comparison of the pose estimation performance. For KCCA, we follow the strategy in [12] and kernelize only the pose parameter space rather than the image space. The RBF kernel function is employed for $\boldsymbol{Y}$ set, i.e. $K(\boldsymbol{y}_1, \boldsymbol{y}_2) = \exp\left(-\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|^2 / 2\sigma^2\right)$, where σ is specified according to the same strategy as that for t_y (see Sec. 3.2), and the optimal result is determined by trial and error. For LPCCA, the local neighbor is defined according to *k*-nearest neighbor, and *k* ranges from 2 to 6, which indicates that for each testing sample, the training images whose corresponding poses deviating approximately ±10 ~±45 degree from that of the testing image are chosen as its local neighbors. The optimal value of *k* is also specified through a trial and error manner. The mean estimation errors (means) and standard deviations (std) using CCA, KCCA and LPCCA are tabulated in Table 1 and 2 when *L* is set to 2 and 3, respectively.

In Table 1, when the pose resolution is approximately 10 degree (*L*=2 case), KCCA only wins on subject 2, 11, 13 and 14, while LPCCA outperforms CCA and KCCA for the rest 16 subjects. In Table 2, when the pose resolution is approximately 15 degree (*L*=3 case), KCCA only wins for subject 2, 3, 11, 13 and 18, while LPCCA outperforms other methods for the rest 15 subjects. Clearly speaking, when *L*=2, for subject 15 and 17, the errors of LPCCA are 1-2 degree lower than those of CCA and KCCA; and for subject 10, the error of KCCA, 0.73 degree, is very precise, rather, it loses to 0.48 degree, the error of LPCCA. When *L*=2 and 3, for subject 9 and 19, large errors and standard deviations appear for CCA and/or KCCA (similar things happen for subject 6 and 14 when *L*=3), however, the errors of LPCCA still remain relatively small, ranging from 2.3 to 6.5 degree. A further analysis of the large errors and the standard deviations for CCA and KCCA indicates that large estimation errors appear on a few testing samples, thus undermining the overall estimation accuracy. In contrast, this

notorious phenomenon is avoided in estimation using LPCCA.

Table 1. The pose estimation accuracies (unit: degree) when L is taken 2

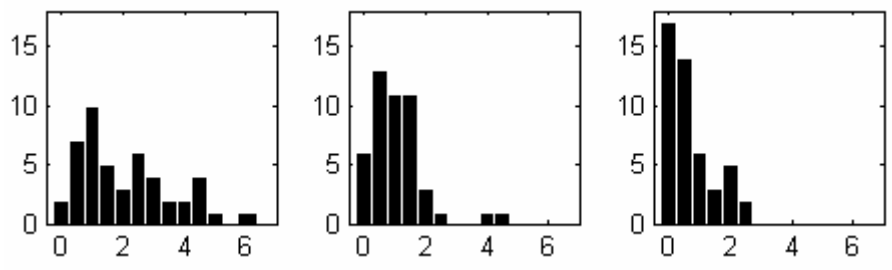
	CCA		KCCA		LPCCA	
subject	mean	std	mean	std	mean	std
1	0.44	0.38	0.55	0.55	0.42	0.36
2	1.78	1.73	1.17	1.17	1.32	1.23
3	3.93	3.31	2.04	2.28	1.84	1.40
4	1.01	0.84	0.90	0.65	0.81	0.74
5	1.87	1.42	1.25	0.98	0.76	0.91
6	3.86	5.72	2.37	4.70	2.10	1.87
7	1.36	1.01	1.35	1.74	0.93	0.81
8	2.12	2.31	1.24	1.38	1.00	1.02
9	6.69	9.28	9.53	41.02	2.32	1.70
10	1.06	1.04	0.73	0.55	0.49	0.37
11	3.22	2.80	1.57	1.21	2.26	1.87
12	1.86	1.76	1.97	2.24	1.52	1.39
13	1.73	1.57	0.91	0.79	0.94	0.98
14	4.02	3.77	1.91	2.35	1.94	1.73
15	2.26	2.38	2.11	1.72	0.62	0.53
16	1.43	1.36	3.46	3.46	1.01	0.66
17	2.54	2.27	3.08	3.35	1.10	1.33
18	2.10	1.77	2.34	2.31	1.64	1.32
19	14.41	57.47	11.85	57.34	3.20	2.36
20	1.75	1.40	1.74	1.74	1.33	1.10

Table 2. The pose estimation accuracies (unit: degree) when L is taken 3

	CCA		KCCA		LPCCA	
subject	mean	std	mean	std	mean	std
1	0.59	0.60	0.65	0.59	0.52	0.57
2	1.78	1.75	1.17	1.19	1.35	1.32
3	4.62	3.60	2.82	4.06	3.51	3.94
4	1.19	1.13	0.95	0.70	0.85	0.82
5	3.17	2.64	2.15	1.42	1.98	1.56
6	15.03	31.77	13.16	41.03	5.86	5.15
7	2.10	1.53	2.40	3.38	1.40	1.01
8	2.61	2.56	1.77	1.79	1.06	1.10
9	12.12	37.21	8.29	35.50	4.57	3.75
10	2.08	1.47	1.10	0.91	0.70	0.71
11	4.07	3.47	2.19	2.13	2.47	2.45
12	2.28	2.22	2.17	2.53	1.99	1.75

13	2.79	2.68	1.27	1.19	2.02	1.87
14	14.83	50.48	18.19	70.78	5.73	7.08
15	2.45	2.26	2.88	2.49	1.41	1.60
16	1.31	1.02	2.78	2.14	1.08	1.07
17	3.14	2.53	4.12	4.11	2.06	1.89
18	4.23	3.78	3.80	3.71	3.83	3.86
19	14.64	50.11	9.66	49.47	6.57	6.95
20	2.18	1.83	2.28	1.90	2.04	1.68

The error-frequency histograms for subject 10 in the case of L=3 are shown in Fig. 9. From Fig.9 we can obtain the following observations: 1) large error (>2 degree) frequently occurs in CCA (Fig. 9a), resulting in large error and standard deviation; 2) such a case becomes a bit better in KCCA ((Fig. 9b), where most of the errors range from 0.5 to 1.5 degree, but large error still appears, e.g. the error near 4 and 4.5 degree appears one time, respectively; and 3) for LPCCA (Fig. 9c), the errors that appear most frequently are near 0 and 0.5 degree (appear for 17 times and 14 times in 47 testing samples), and the maximal error is not more than 2.5 degree, which results in a rather small error and standard deviation, namely, 0.70±0.71 degree. This comparison between the distributions of errors reveals more robust behavior brought by LPCCA.



(a) (b) (c)

Fig. 9. The error-frequency histograms for subject 10 and L=3 case respectively using (a) CCA, (b) KCCA and (c) LPCCA. The horizontal coordinate denotes estimation error in degree and the vertical coordinate denotes the frequency of occurrence.

The parametric manifolds for pose estimation obtained by CCA, KCCA and LPCCA are respectively given in Fig. 10 to vividly illustrate the distributions of the estimation errors in Fig. 9. From Fig. 10 we can observe that 1) in the parametric manifold of CCA (Fig. 10a), the obvious deviation of testing samples from the overall parametric manifold occurs in some

patches; and the profile of the manifold looks like a simple circle; 2) such a deviation also occurs in KCCA (Fig. 10b); and the overall profile of the manifold is a bit more complex closed curve; and 3) the deviation mentioned above seems to be confined to some limited extent in the parametric manifold of LPCCA (Fig. 10c); and the overall profile of the manifold seems to be an irregular curve and more complex than that of KCCA (Fig. 10b). Small deviations will benefit to the stability of pose estimation, whereas the large deviation will increase the risk of the unstable prediction. Since the nature of pose estimation using all these methods is curve fitting and prediction, the large deviation should be avoided as much as possible to improve the estimation accuracy. It seems that the construction of the local manifold confines the large deviation obviously, just as fitting a complex, unknown function using piecewise low-order smooth curve to avoid undermining the generalization ability. So we can owe the improvement of the estimation performance to the following facts: 1) the introduction of the local structure of the data to CCA enables the dimensionality reduction of LPCCA to preserve the intrinsic local characteristics of the data; and 2) the construction of the local parametric manifold constrains the large deviation mentioned above.

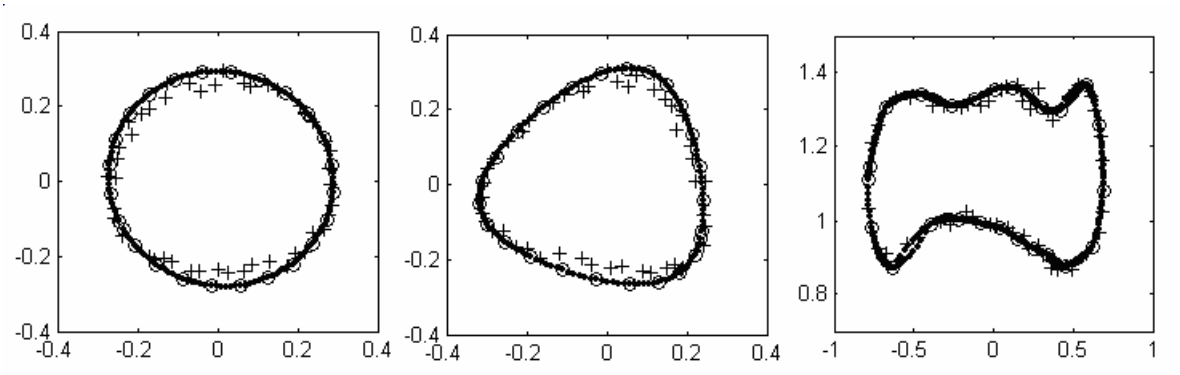


Fig. 10. The parametric manifolds for pose estimation obtained respectively by (a) CCA, (b) KCCA and (c) LPCCA. Circle denotes training data, and black dot the resampling data through cubic spline interpolation [12,40]. All these discrete data points construct the parametric manifold of a subject. The cross + denotes testing data. The overlapping and junction appear in (c) due to the conjunction of the local manifold.

6. Conclusion and future work

When high dimensional data lies in a low dimensional manifold embedded in the ambient space, CCA and KCCA can discover the manifold structure to some extent. However, both of them are globality based approach and the local structure information hidden in the data is

not taken into account. When some complex, nonlinear manifolds are encountered, CCA and KCCA will be challenged and even failed. To attack this problem, in this paper, the local structure information is introduced into CCA, so the globally nonlinear problem is decomposed into a series of locally linear sub-problems, and reversely the optimization solution to the combination of these locally linear sub-problems gives rise to basis vectors for dimensionality reduction. The experiment of data visualization suggests that the proposed method, namely LPCCA, generalize the traditional CCA, such that it can not only capture the canonical correlation between the data pairs but also preserve the local structure of the data to a better extent. The experiment of pose estimation on COIL-20 dataset suggests that LPCCA outperforms both CCA and KCCA in pose estimation.

Up to now, KCCA and its variants have been applied to pose estimation [12], pattern recognition [9,10], image retrieval [8], text analysis and retrieval [14,15], bioinformatics [19] and other fields. As another nonlinear extension of CCA, LPCCA will play its necessary role in these fields.

Furthermore, compared with KCCA which is globally nonlinear, although LPCCA can deal with the problem which is globally nonlinear but locally linear, it is linear dimensionality reduction in nature so that it is possible to kernelize this method to match the rigorous requirement of some nonlinear problem in real world. In fact the kernel version using linear kernel is formally given in (16). This is just our current research topic.

Acknowledgement

We would like thank the anonymous reviewers' constructive suggestion for greatly improving the presentation of this paper and Natural Science Foundations of Jiangsu under grant No. BK2005122 for support, and we would like to thank Michael Reiter and Thomas Melzer very much for offering their source codes generously.

Appendix: proof of the *Theorem* 1 (similar to [12]).

Proof: Eq.(15) can be rewritten as the form of $\boldsymbol{Aw} = \lambda \boldsymbol{Bw}$, where $\boldsymbol{A}$, $\boldsymbol{B}$ are both symmetric matrices of size $(p+q)\times(p+q)$, so $\boldsymbol{B}$ can be eigen-decomposed into $\boldsymbol{B} = \boldsymbol{E\Theta E}^T$, where $\boldsymbol{E}$ is a

orthogonal matrix of size $(p+q)\times(p+q)$ so it satisfies $\boldsymbol{E}^{-1}=\boldsymbol{E}^T$, $\boldsymbol{\Theta}$ a diagonal one with the diagonal entries τ_i, $i=1,\ldots, p+q$. The ith column of $\boldsymbol{E}$, $\boldsymbol{e}_i$, satisfies $\boldsymbol{B}\boldsymbol{e}_i=\tau_i\boldsymbol{e}_i$, that is

$$\boldsymbol{B}\boldsymbol{e}_i=\begin{pmatrix}\boldsymbol{X}\boldsymbol{S}_{xx}\boldsymbol{X}^T & \\ & \boldsymbol{Y}\boldsymbol{S}_{yy}\boldsymbol{Y}^T\end{pmatrix}\begin{pmatrix}\boldsymbol{u}_i\\ \boldsymbol{v}_i\end{pmatrix}=\tau_i\begin{pmatrix}\boldsymbol{u}_i\\ \boldsymbol{v}_i\end{pmatrix},\ i=1,2\ldots p+q. \qquad \text{(A.1)}$$

where $\boldsymbol{e}_i\in \mathrm{R}^{p+q}$ is partitioned into $\boldsymbol{u}_i\in \mathrm{R}^p$ and $\boldsymbol{v}_i\in \mathrm{R}^q$, i.e., let $\boldsymbol{e}_i^T=\left[\boldsymbol{u}_i^T, \boldsymbol{v}_i^T\right]$.

(A.1) can be rewritten as

$$\begin{pmatrix}\boldsymbol{u}_i\\ \boldsymbol{v}_i\end{pmatrix}=\begin{pmatrix}\boldsymbol{X}\cdot\boldsymbol{S}_{xx}\boldsymbol{X}^T\boldsymbol{u}_i/\tau_i\\ \boldsymbol{Y}\cdot\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{v}_i/\tau_i\end{pmatrix}=\begin{pmatrix}\boldsymbol{X}\boldsymbol{\xi}_i\\ \boldsymbol{Y}\boldsymbol{\zeta}_i\end{pmatrix},\ i=1,2\ldots p+q. \qquad \text{(A.2)}$$

where $\boldsymbol{\xi}_i=\boldsymbol{S}_{xx}\boldsymbol{X}^T\boldsymbol{u}_i/\tau_i\in R^n$ and $\boldsymbol{\zeta}_i=\boldsymbol{S}_{yy}\boldsymbol{Y}^T\boldsymbol{v}_i/\tau_i\in R^n$. So we have

$$\boldsymbol{u}_i=\boldsymbol{X}\boldsymbol{\xi}_i,\ \boldsymbol{v}_i=\boldsymbol{Y}\boldsymbol{\zeta}_i \qquad \text{(A.3)}$$

which shows that $\boldsymbol{u}_i$, $\boldsymbol{v}_i$ are linear combinations of the training data.

Substituting $\boldsymbol{B}^{-1}=\boldsymbol{E}\boldsymbol{\Theta}^{-1}\boldsymbol{E}^T$ into $\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{w}=\lambda\boldsymbol{w}$ and obtaining

$$\boldsymbol{E}\cdot\boldsymbol{\Theta}^{-1}\boldsymbol{E}^T\boldsymbol{A}\boldsymbol{w}=\lambda\boldsymbol{w} \qquad \text{(A.4)}$$

where $\boldsymbol{\Theta}^{-1}\boldsymbol{E}^T\boldsymbol{A}\boldsymbol{w}$ is a $(p+q)$-dimensional vector and can be expressed as $\left(c_1,\ldots,c_{p+q}\right)^T$, then (A.4) becomes

$$\begin{pmatrix}\boldsymbol{u}_1\ldots\boldsymbol{u}_{p+q}\\ \boldsymbol{v}_1\ldots\boldsymbol{v}_{p+q}\end{pmatrix}\begin{pmatrix}c_1\\ \ldots\\ c_{p+q}\end{pmatrix}=\lambda\begin{pmatrix}\boldsymbol{w}_x\\ \boldsymbol{w}_y\end{pmatrix} \qquad \text{(A.5)}$$

where the vector $\boldsymbol{w}\in \mathrm{R}^{p+q}$ is partitioned into $\boldsymbol{w}_x\in \mathrm{R}^p$ and $\boldsymbol{w}_y\in \mathrm{R}^q$, i.e., let $\boldsymbol{w}^T=\left[\boldsymbol{w}_x^T, \boldsymbol{w}_y^T\right]$. So we have

$$\boldsymbol{w}_x=\frac{1}{\lambda}\sum_{i=1}^{p+q}c_i\boldsymbol{u}_i=\frac{1}{\lambda}\sum_{i=1}^{p+q}c_i\boldsymbol{X}\boldsymbol{\xi}_i=\boldsymbol{X}\cdot\frac{1}{\lambda}\sum_{i=1}^{p+q}c_i\boldsymbol{\xi}_i=\boldsymbol{X}\alpha \qquad \text{(A.6)}$$

where $\boldsymbol{\alpha}=\frac{1}{\lambda}\sum_{i=1}^{p+q}c_i\boldsymbol{\xi}_i\in R^n$. Due to the similar deduction, we have $\boldsymbol{w}_y=\boldsymbol{Y}\boldsymbol{\beta}$, so *Theorem* 1 holds.

If $\boldsymbol{B}$, however, is singular, we can perform eigen-decomposition on matrix $\boldsymbol{B}+\mu\boldsymbol{I}$ rather on $\boldsymbol{B}$, where μ is a regularization parameter. Note that this operation simply shifts the eigenvalues of $\boldsymbol{B}$ but leaves its eigenvectors unchanged, so *Theorem* 1 still holds.

References

[1] M. Borga, Canonical correlation: a tutorial, at http://people.imt.liu.se/~magnus/cca/tutorial/, 1999

[2] H. Hotelling, Relations between two sets of variates, Biometrika, 28(1936), 321-377.

[3] Yacov Hel-Or, The canonical correlations of color images and their use for demosaicing, HP Labs Technical Report, HPL-2003-164(R.1), Feb. 2004

[4] M. Loog, B. van Ginneken, R. P.W. Duin, Dimensionality reduction of image features using the canonical contextual correlation projection, Pattern Recognition, 38(2005), 2409-2418.

[5] A.A. Nielsen, Multiset Canonical correlations analysis and multispectral, truly multitemporal remote sensing data, IEEE Transactions on Image Processing 11(2002), 293-305.

[6] M. Borga, Learning Multidimensional signal processing, ph.D thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden,1998.

[7] O. Friman, M, Borga, P. Lundberg, H. Knutsson, Canonical correlation as a tool in functional MRI data analysis, SSAB 2001, Proceedings of the SSAB Symposium on Image Analysis, March, Norrköping, Sweden, 2001

[8] D.R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Computation, 16(2004), 2639-2664.

[9] T.V.Gestel, J.A.K.Suykens, J. De Brabanter, B. De Moor, J.Vandewalle, Kernel canonical correlation analysis and least squares support vector machines，Proc. of the International Conference on Artificial Neural Networks (ICANN 2001) 384-389.

[10] Yo Horikawa, Use of autocorrelation kernels in kernel canonical correlation analysis for texture classification, in N.R. Pal et al. (Eds.): ICONIP 2004, LNCS 3316, pp. 1235.1240, 2004. Springer-Verlag Berlin Heidelberg

[11] Quan-Sen Sun, Sheng-Gen Zeng, Yan Liu, Pheng-Ann Heng, De-Shen Xia, A new method of feature fusion and its application in image recognition, Pattern Recognition, 38(2005), 2437-2448.

[12] T.Melzer, M.Reiter, H.Bischof, Appearance models based on kernel canonical correlation analyusis, Pattern Recognition, 36(2003), 1961-1971.

[13] E. Kidron, Y.Y. Schechner, M. Elad, Pixels that Sound, IEEE Proc. of Computer Vision and Pattern Recognition, 1, 88-95, June 2005

[14] Blza Fortuna, Kernel canonical correlation analysis with applications, In: SIKDD 2004 at

Multiconference IS 2004, 12-15 Oct 2004, Ljubljana, Slovenia.

[15] J. Shawe-Taylor, N. Cristianini, Kernel methods for pattern analysis, Cambridge Press, U.K., 2004

[16] B. Abraham, G. Merola, Dimensionality reduction approach to multivariate prediction, Computational Statistics & Data Analysis, 48(2005), 5-16.

[17] N. Vlassis, Y. Motomura, B. Krose, Supervised linear feature extraction for mobile robot localization, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'00), San Francisco, CA, 2000, pp. 2979-2984

[18] H. Pan, Zhi-Pei Liang, T.S. Huang, Exploiting the dependencies in information fusion, IEEE Proc. of the Conf. on Computer Vision and Pattern Recognition, 1999, 2, pp.407-412

[19] Y. Yamanishi, J.P Vert, A. Nakaya, M. Kanehisa , Extraction of correlated gene clusters from multiple genomic data by generalized kernel canonical correlation analysis, Bioinformatics, 19(2003), i323-330

[20] Mathworks Inc., Matlab 7.0 Release 14 help: Statistics toolbox, Jan. 2005

[21] N. Cristianini, J. Shawe-Taylor, An Introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, 2000.

[22] P.L. Lai, Neural implementations of canonical correlation analysis, ph.D thesis, Dept. of computing and information systems, University of Paisley, Scotland, March 2000.

[23] Zhenkun Gou, Colin Fyfe, A canonical correlation neural network for multicollinearity and functional data, Neural Networks 17 (2004) 285–293

[24] J.B.Tenenbaum, Vin de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science, 290(2000), 2319-2323.

[25] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science, 290(2000), 2323-2326.

[26] X. He, P. Niyogi, Locality preserving projections, Advances in Neural Information Processing Systems, 2004

[27] X. He, S.Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using Laplicianfaces, IEEE Trans. Pattern Anal. and Machine Intelli. 27 (2005), 328-340.

[28] W. Min, K. Lu, X. He, Locality pursuit embedding, Pattern Recognition, 37(2004), 781-788.

[29] N. Kambhatla, T. K. Leen, Dimension reduction by local principal component analysis. Neural

Computation, 9(1997), 1493-1516.

[30] J.J Verbeek, S.T. Roweis, N. Vlassis, Non-linear CCA and PCA by alignment of local models, Advances in Neural Information Processing Systems, 2004

[31] Tae-Kyun Kim, J. Kittler, Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image, IEEE Trans. Pattern Analysis and Machine Intelligence, 27(2005) :318 – 327.

[32] N.E. Ayat, M. Cheriet, C.Y. Suen, Automatic model selection for the optimization of SVM kernels, Pattern Recognition, 38(2005), 1733 – 1745

[33] E. Parrado-Hernández, J. Arenas-García, I. Mora-Jiménez, A. Navia-Vázquez, On problem-oriented kernel refining, Neurocomputing, 55(2003), 135-150.

[34] J.H. Friedman, Regularized discriminant analysis, Journal of the American Statistics Association, 84(1989), 165-175.

[35] S. Mika, G. Ratsch, J. Weston, B. Scholköpf, K.-R. Müller, Fisher discriminant analysis with kernels, in Neural Networks for Signal Processing IX, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds., pp. 41-48. IEEE Press, New York, 1999

[36] L. K. Saul, S.T. Roweis, Think globally, fit locally: unsupervised learning of low dimensional manifolds, Journal of Machine Learning Research 4 (2003) 119-155

[37]. L. Hoegaerts, J.A.K. Suykens, J. Vandewalle, B. De Moor, Subset based least squares subspace regression in RKHS. Neurocomputing, 63(2005), 293-323.

[38] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Proc. Conf. Advances in Neural Information Processing System 15, 2001.

[39] H.S. Seung, D.Lee, The manifold ways of perception, Science, 290(2000), 2268-2269.

[40] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, International J. Comp. Vision, 14(1995), 5-24.

[41] S.K. Nayar, S.A. Nene, H. Murase, Subspace methods for robot vision, IEEE Trans. Robotics and Automation,12(1996), 750-758.

[42] B. Raytchev, I. Yoda, K. Sakaue, Head pose estimation by nonlinear manifold learning, IEEE Proc. of the 17th International Conf. on Pattern Recognition, 2004.

[43] H. Murase, S.K. Nayar, Illumination planning for object recognition using parametric eigenspace, IEEE Trans. Pattern Analysis and Machine Intelligence, 16(1994), 1219-1227.

[44] S. A. Nene, S. K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Technical Report CUCS-005-96, February 1996

[45] B. Schölkopf, A. Smola, K-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation, 10(1998), 1299-1319.