Nonlinear Feature Extraction Using Generalized Canonical Correlation Analysis

Thomas Melzer, Michael Reiter, and Horst Bischof*

The authors are with the Pattern Recognition and Image Processing Group, Vienna University of Technology, Vienna, Austria
{melzer,rei,bis}@prip.tuwien.ac.at

Abstract. This paper introduces a new non-linear feature extraction technique based on *Canonical Correlation Analysis* (CCA) with applications in regression and object recognition. The non-linear transformation of the input data is performed using kernel-methods. Although, in this respect, our approach is similar to other *generalized* linear methods like kernel-PCA, our method is especially well suited for relating two sets of measurements. The benefits of our method compared to standard feature extraction methods based on PCA will be illustrated with several experiments from the field of object recognition and pose estimation.

1 Introduction

When dealing with high-dimensional observations, linear mappings are often used to reduce the dimensionality of the data by extracting a small (compared to the superficial dimensionality of the data) number of linear features, thus alleviating subsequent computations. A prominent example of a linear feature extractor is *Principal Component Analysis* (PCA [4]). Among all linear, orthonormal transformations, PCA is optimal in the sense that it minimizes, in the mean square sense, the reconstruction error between the original signal $\mathbf{x}$ and the signal $\hat{\mathbf{x}}$ reconstructed from its low-dimensional representation $\mathbf{f}(\mathbf{x})$. During the recent years, PCA has been especially popular in the object recognition community, where it has successfully been employed in various applications such as face recognition [13], illumination planning [9], visual inspection and even visual servoing [11].

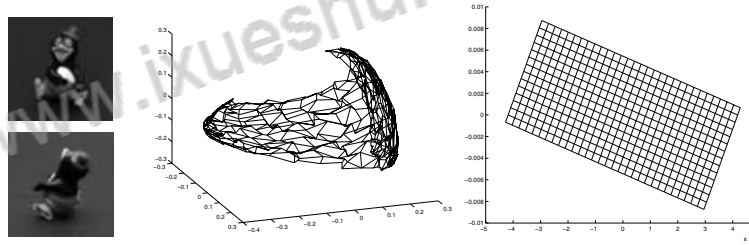
Although this demonstrates the broad applicability of PCA, one has to bear in mind that the goal of PCA is minimization of the reconstruction error; in particular, PCA-features are not well suited for regression tasks. Consider a mapping $\phi : \mathbf{x} \mapsto \mathbf{y}$. There is no reason to believe that the features extracted by PCA on the variable $\mathbf{x}$ will reflect the functional relation between $\mathbf{x}$ and $\mathbf{y}$ in any way; even worse, it is possible that information vital to establishing this relation is discarded when projecting the original data onto the PCA-feature space.

* This work was supported by the Austrian Science Foundation (FWF) under grant no. P13981-INF.

G. Dorffner, H. Bischof, and K. Hornik (Eds.): ICANN 2001, LNCS 2130, pp. 353–360, 2001.

There exist, however, several other linear methods that are better suited for regression tasks, for example *Partial Least Squares* (PLS [3]), *Multivariate Linear Regression* (MLR, also referred to as *Reduced Rank Wiener Filtering*, see for example [2]) and *Canonical Correlation Analysis* (CCA [5]). Among these three, only MLR gives a direct solution to the linear regression problem. PLS and CCA will find pairs of directions that yield maximum covariance resp. maximum correlation between the two random variables $\mathbf{x}, \mathbf{y}$; regression can then be performed on these features. CCA, in particular, has some very attractive properties (for example, it is invariant w.r.t. affine transformations - and thus scaling - of the input variables) and can not only be used for regression purposes, but whenever we need to establish a relation between two sets of measurements (e.g., finding corresponding points in stereo images [1]).

As an example for CCA, consider constructing a parametric manifold for pose estimation [10]. Fig. 1(a) shows two extreme views of an object, which was acquired with two varying pose parameters (pan and tilt). Let $\mathbf{X}$ denote the set of training images and $\mathbf{Y}$ the set of corresponding pose parameters.The visualization of the manifold given in Fig. 1(b) is obtained by plotting the projections of the training set onto the first three eigenvectors obtained by standard PCA, whereby neighboring (w.r.t. the pose parameters) projections are connected.



(a) (b) (c)

Fig. 1. Extreme views of training set (a) and the parametric manifold obtained with PCA (b) and CCA (c).

The parametric manifold serves as a starting point for computing pose estimates for new input images. The standard-approach for retrieving these estimates is to resample the manifold using, e.g., bicubic spline interpolation and then to perform a nearest neighbor search for each new image [10].

Fig. 1(c) shows the manifold obtained by projecting the training images onto the first two directions found by computing CCA on $\mathbf{X}$ and $\mathbf{Y}$ (the number of factors obtained by CCA is limited by the dimensionality of the lower-dimensional set). In contrast to the PCA-manifold, the CCA-factors span a perfect grid; one could also say that projections of the training images onto the two linear features found by CCA are topologically ordered w.r.t. their associated pose parameters. It is obvious that pose estimation on the manifold obtained by CCA is much easier than on the PCA-manifold.

Recently, we proposed a non-linear extension of CCA by the use of kernel-methods [7]. Kernel-methods have become increasingly popular during the last few years, and have already been applied to PCA [12] and the *Fisher Discriminant* [8]. In our derivation of kernel-CCA we have used the fact that the solutions (principal directions) of CCA can be obtained as the extremum points of an appropriately chosen *Rayleigh Quotient* (this is also true for the other linear techniques discussed thus far, see [1]).

In this paper we will demonstrate the benefits of kernel-CCA with an application in the field of appearance-based pose estimation. To this end we will compare the performance of features obtained by PCA, standard CCA and kernel-CCA.

The rest of this paper is organized as follows: in section 2, we will give a brief introduction to "classical" CCA and show how kernel-methods can be applied to CCA. In section 3 we will apply kernel-CCA for pose estimation. Conclusions will be given in section 4.

2 Canonical Correlation Analysis (CCA)

2.1 What Is CCA?

Given two zero-mean random variables $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$, CCA finds pairs of directions $\mathbf{w}_x$ and $\mathbf{w}_y$ that maximize the correlation between the projections $x = \mathbf{w}_x^T\mathbf{x}$ and $y = \mathbf{w}_y^T\mathbf{y}$ (in the context of CCA, the projections x and y are also referred to as *canonical variates*). More formally, CCA maximizes the function:

$$\rho = \frac{E[xy]}{\sqrt{E[x^2]E[y^2]}} = \frac{E[\mathbf{w}_x^T\mathbf{x}\mathbf{y}^T\mathbf{w}_y]}{\sqrt{E[\mathbf{w}_x^T\mathbf{x}\mathbf{x}^T\mathbf{w}_x]E[\mathbf{w}_y^T\mathbf{y}\mathbf{y}^T\mathbf{w}_y]}}, \tag{1}$$

$$\rho = \frac{\mathbf{w}_x^T\mathbf{C}_{xy}\mathbf{w}_y}{\sqrt{\mathbf{w}_x^T\mathbf{C}_{xx}\mathbf{w}_x\mathbf{w}_y^T\mathbf{C}_{xy}\mathbf{w}_y}}. \tag{2}$$

Let

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{0} \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{pmatrix}. \tag{3}$$

It can be shown [1] that the stationary points $\mathbf{w}^* = (\mathbf{w}_x^{*T}, \mathbf{w}_y^{*T})^T$ of ρ (i.e., the points satisfying $\nabla\rho(\mathbf{w}^*) = \mathbf{0}$) coincide with the stationary points of the *Rayleigh Quotient*:

$$r = \frac{\mathbf{w}^T\mathbf{A}\mathbf{w}}{\mathbf{w}^T\mathbf{B}\mathbf{w}}, \tag{4}$$

and thus, by virtue of the *Generalized Spectral Theorem* [2], can be obtained as solutions (i.e., eigenvectors) of the corresponding generalized eigenproblem:

$$\mathbf{A}\mathbf{w} = \mu\mathbf{B}\mathbf{w}. \tag{5}$$

The extremum values $\rho(\mathbf{w}^*)$, which are referred to as *canonical correlations*, are equally obtained as the corresponding extremum values of Eq. 4 or the eigenvalues of Eq. 5, respectively, i.e., $\rho(\mathbf{w}^*) = r(\mathbf{w}^*) = \mu(\mathbf{w}^*)$.

2.2 Kernel CCA

In this section we will briefly summarize the formulation of kernel-CCA, which can be used to find non-linear dependencies between two sets of observations. A detailed derivation of the algorithm can be found in [6].

Given n pairs of mean-normalized observations $(\mathbf{x}_i^T, \mathbf{y}_i^T)^T \in \mathbb{R}^{p+q}$, and data matrices $\mathbf{X} = (\mathbf{x}_1..\mathbf{x}_n) \in \mathbb{R}^{p\times n}$, $\mathbf{Y} = (\mathbf{y}_1..\mathbf{y}_n) \in \mathbb{R}^{q\times n}$, we obtain the estimates for the covariance matrices $\mathbf{A}, \mathbf{B}$ in Eq. 3 as

$$\hat{\mathbf{A}} = \frac{1}{n}\begin{pmatrix} \mathbf{0} & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & \mathbf{0} \end{pmatrix},\ \hat{\mathbf{B}} = \frac{1}{n}\begin{pmatrix} \mathbf{X}\mathbf{X}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}\mathbf{Y}^T \end{pmatrix} \tag{6}$$

If the mean was estimated from the data, we have to replace n by $n-1$ in both equations.

As we know from section 2.1, computing the CCA between the data sets $\mathbf{X}, \mathbf{Y}$ amounts to determining the extremum points of the *Rayleigh Quotient* (see Eq. 4). It can be shown [6], that for all solutions $\mathbf{w}^* = (\mathbf{w}_x^{*T}, \mathbf{w}_y^{*T})^T$ of Eq. 5, the component vectors $\mathbf{w}_x^*, \mathbf{w}_y^*$ lie in the span of the training data (i.e., $\mathbf{w}_x^* \in span(\mathbf{X})$ and $\mathbf{w}_y^* \in span(\mathbf{Y})$). Under this assumption for each eigenvector $\mathbf{w}^* = (\mathbf{w}_x^{*T}, \mathbf{w}_y^{*T})^T$ solving Eq. 5, there exist vectors $\mathbf{f}, \mathbf{g} \in \mathbb{R}^n$, so that $\mathbf{w}_x^* = \mathbf{X}f$ and $\mathbf{w}_y^* = \mathbf{Y}g$. Thus, CCA can completely be expressed in terms of dot products. This allows us to reformulate the *Rayleigh Quotient* using only inner products:

$$\frac{\begin{pmatrix}\mathbf{f}^T & \mathbf{g}^T\end{pmatrix}\begin{pmatrix} \mathbf{0} & \mathbf{K}\mathbf{L} \\ \mathbf{L}\mathbf{K} & \mathbf{0} \end{pmatrix}\begin{pmatrix}\mathbf{f} \\ \mathbf{g}\end{pmatrix}}{\begin{pmatrix}\mathbf{f}^T & \mathbf{g}^T\end{pmatrix}\begin{pmatrix} \mathbf{K}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{L}^2 \end{pmatrix}\begin{pmatrix}\mathbf{f} \\ \mathbf{g}\end{pmatrix}}, \tag{7}$$

where $\mathbf{K}, \mathbf{L}$ are Gram matrices defined by $\mathbf{K}_{ij} = \mathbf{x}_i^T\mathbf{x}_j$ and $\mathbf{L}_{ij} = \mathbf{y}_i^T\mathbf{y}_j$, $\mathbf{K}, \mathbf{L} \in \mathbb{R}^{n\times n}$. The new formulation makes it possible to compute CCA on non-linearly mapped data without actually having to compute the mapping itself. This can be done by substituting the gram matrices $\mathbf{K}, \mathbf{L}$ by kernel matrices $\mathbf{K}^\phi{}_{ij} = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j) = k_\phi(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{L}^\theta{}_{ij} = \theta(\mathbf{y}_i)^T\theta(\mathbf{y}_j) = k_\theta(\mathbf{y}_i, \mathbf{y}_j)$. $k_\phi(.,.), k_\theta(.,.)$ are the kernel functions corresponding to the nonlinear mappings $\phi : \mathbb{R}^p \to \mathbb{R}^{m_1}$ resp. $\theta : \mathbb{R}^q \to \mathbb{R}^{m_2}$.

The projections onto $\mathbf{w}_\phi^*$ can be computed using only the kernel function, without having to evaluate ϕ itself:

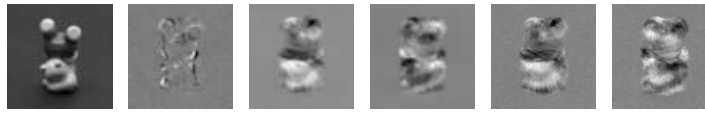
$$\phi(\mathbf{x})^T\mathbf{w}_\phi^* = \sum_{i=1}^{n} f_{\phi i}\phi(\mathbf{x})^T\phi(\mathbf{x}_i) = \sum_{i=1}^{n} f_{\phi i}k(\mathbf{x}, \mathbf{x}_i). \tag{8}$$

The projections onto $\mathbf{w}_\theta^*$ are obtained analogously.

Note that using kernel-CCA, we can compute more than $min(p, q)$ (p, q being the dimensionality of the variable $\mathbf{x}$ and $\mathbf{y}$, respectively) factor pairs, which is the limit imposed by classical CCA.

3 Experiments

In the following example we apply CCA to a pose estimation problem, where we relate images of objects at varying pose to corresponding pose parameters. Experiments were conducted on three test objects, detailed quantitative figures are given only for object 2(a). However, results for the remaining objects are similar.



(a) (b) (c) (d) (e) (f)

Fig. 2. (a) Image $\mathbf{x}_1$ of one of our test objects. (b) Canonical factor $\mathbf{w}_x^*$ computed on $\mathbf{X}_8^t$. (c),(d) Canonical factors obtained on the same set using a non-linear (trigonometric) representation of the output parameter space. (e),(f) 2 factors obtained by kernel-CCA with canonical correlations $\rho = 1.0$.

Let $\mathbf{X} = \langle \mathbf{x}_i | 1 \leq i \leq 180 \rangle$ denote the set of images and $\mathbf{Y} = \langle y_i | y_i \in [0, 358] \rangle$ corresponding pose parameters (horizontal orientation of the object w.r.t. the camera in degrees). The images are represented as 128^2-dimensional vectors that are obtained by sequentially stacking the image columns.

Each image set $\mathbf{X}$ was subsampled to obtain a subset of images that was used as a training set. The remaining images were assigned to the corresponding test set. Let $\mathbf{X}_k$ denote a training set that was generated by subsampling $\mathbf{X}$ at every kth position[1]. Since y_{i_k} is a scalar, standard CCA yields only one feature vector (canonical factor) $\mathbf{w}_x^*$ (see figure 2(b)). Figure 4 (in the upper 2 plots) gives a quantitative comparison of the pose estimation error ϵ (linear regression model) using the PCA and standard CCA. The pose estimation error ϵ was calculated on the test set as the absolute difference between known and estimated parameter value (orientation in degrees).

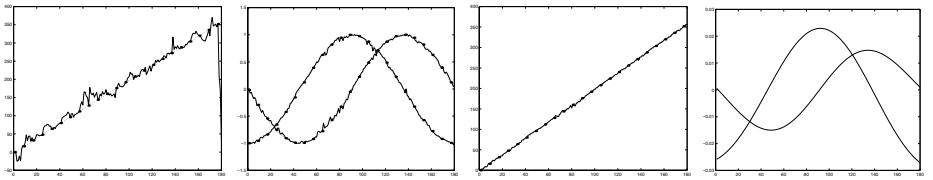
Figure 3(a) shows a plot of pose estimates for the complete set of images $\mathbf{X}$ obtained using CCA. The dotted line indicates the true pose parameter values y_i. Pose parameter values of $\mathbf{X}_k$ are marked by filled circles. In figure 3 a linear least-square mapping was used to map the projections of the training set to pose parameters.

Note that the pose estimation error grows rapidly around $i = 180$. This problem is due to the fact that the scalar representation for y_i has a discontinuity at $y_i = 360$. From figure 2(a) it can be seen that the main part of information held in $\mathbf{w}_x^*$ is about the transition of image $\mathbf{x}_{180}$ to image $\mathbf{x}_1$.

For this reason we chose an periodic, trigonometric representation of pose parameters $\mathbf{y}_i = [sin(y_i), cos(y_i)]^T$. The object's pose is now characterized by 2

[1] $\mathbf{X}_k$ contains all images $\mathbf{x}_{i_k}$ with $i_k \in \{1, k, 2k, \ldots, \lfloor 180/k \rfloor\}$. Since the original set shows the object in 2 degrees steps, $\mathbf{X}_k$ shows the object at $2k$ degrees steps.

parameters, and thus CCA yields 2 canonical factors (see figure 2 (c) and (d)) in the image vector space. Estimates obtained for these intermediate output values (again using linear regression) are given in 3(b), while the final pose estimates (obtained by combining these two estimates using atan2) are given in figure 3(c). Thus, by using *a priori* knowledge about the problem domain a significant increase in pose estimation accuracy could be obtained.



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

Fig. 3. Output parameter estimates obtained from feature projections by the linear regression function of the training set. Horizontal axes correspond to the image indices, vertical axes to estimated output parameter values. The dotted line indicates the true pose parameter values. Parameter values of the training set are marked by filled circles. (a) shows estimates using scalar representation of orientation (b) using trigonometric representation. (c) Estimated pose obtained from trigonometric representation using the four-quadrant arc tangens. Note that the accuracy of estimated pose parameters can be improved considerably. (d) Projections onto factors obtained using kernel-CCA: optimal factors can be obtained automatically.

The last experiment shows the relative performance of CCA, kernel-CCA and PCA when using spline interpolation and resampling [10] for pose estimation. For standard CCA we used the hard-coded trigonometric pose representation (yielding 2 factor pairs), while kernel-CCA (using a RBF-kernel with $\sigma = 1.6$) obtained similar factors automatically from the original scalar pose representation (see figures 2 (e)-(f) and 3(d)). Results are given in figure 4 (middle row). The lowermost two plots of figure 4 show results for PCA. The pose estimation error is significantly larger compared to CCA when using the same number of features.

4 Conclusion and Outlook

Although little known in the field of pattern recognition and signal processing, CCA is a very powerful and versatile statistical tool that is especially well suited for relating two sets of measurements. CCA, like PCA, can also be regarded as a linear feature extractor. CCA-features are, however, much better suited for regression tasks than features obtained by PCA; this was demonstrated in section 1 for the task of computing a parametric object manifold for pose estimation.

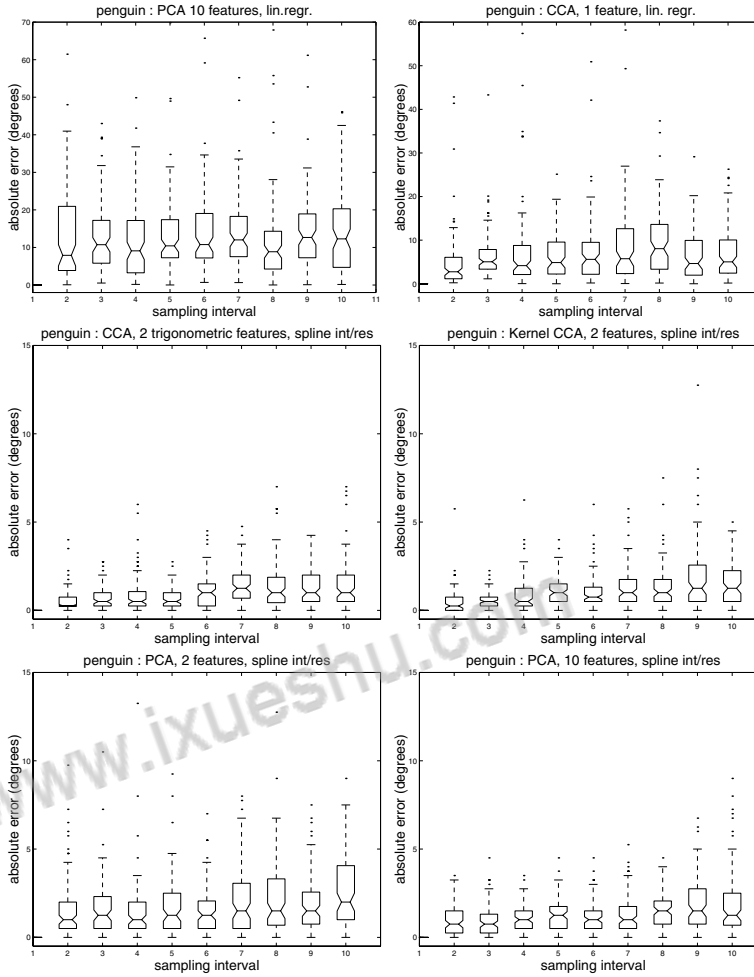


Fig. 4. Uppermost two plots: Comparison of pose estimation error when using a linear regression from factor projections to pose parameters. The left plot shows the errors for a 10 dimensional PCA-feature space. The right plot shows the results when using only one CCA-feature. Image sets have been subsampled at different intervals (horizontal axis) to obtain increasingly smaller training sets. Subsampling was done at at 9 different sampling intervals ($k = \langle 2, ..., 10 \rangle$). Note that the plots have been scaled differently.
Plots in the middle row: Pose estimation error for CCA-features. The left column shows results for "classical" CCA when using a 2 dimensional trigonometric representation of output parameters. The right column shows results when using 2 features obtained by kernel-CCA and a scalar pose representation.
Lower two plots: Pose estimation error using the first 10 eigenvectors (left plot) and only 2 eigenvectors. (right plot). Estimates were obtained from feature projections using spline interpolation and resampling.

In section 2, we discussed how to non-linearly extend CCA by using kernel-functions. **Kernel-CCA** is a efficient non-linear feature extractor, which also overcomes some of the limitations of classical CCA.

Finally, in section 3, we applied kernel-CCA to an object pose estimation problem. There, it was also shown that kernel-CCA will automatically find an optimal, periodic representation for a training set containing object views ranging from 0 to 360 degrees (i.e., for *periodic* data).

Currently, we are investigating methods for obtaining the desired pose parameters (or, in general, output estimates) from the projections onto the transformed output basis vectors $\mathbf{w}_\phi$, i.e., for inverting a kernelized output representation.

References

1. Magnus Borga. *Learning Multidimensional Signal Processing.* Linköping Studies in Science and Technology, Dissertations, No. 531. Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1998.
2. Konstantinos I. Diamantaras and S.Y. Kung. *Principal Component Neural Networks.* John Wiley & Sons, 1996.
3. A. Höskuldsson. PLS regression methods. *Journal of Chemometrics*, 2:211–228, 1988.
4. H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520, 1933.
5. H. Hotelling. Relations between two sets of variates. *Biometrika*, 8:321–377, 1936.
6. Thomas Melzer, Michael Reiter, and Horst Bischof. Kernel CCA: A nonlinear extension of canonical correlation analysis. *Submitted to IEEE Trans. Neural Networks*, 2001.
7. Thomas Melzer and Micheal Reiter. Pose estimation using parametric stereo eigenspaces. In Tomas Svoboda, editor, *Czech Pattern Recognition Workshop*, pages 77–80. Czech Pattern Recognition Society Praha, 2000.
8. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing*, volume 9, pages 41–48. IEEE, 1999.
9. Hiroshi. Murase and Shree K. Nayar. Illumination planning for object recognition using parametric eigenspaces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(12):1219–1227, December 1994.
10. Hiroshi Murase and Shree K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.
11. Shree K. Nayar, Sameer A. Nene, and Hiroshi Murase. Subspace methods for robot vision. *IEEE Trans. Robotics and Automation*, 12(5):750–758, October 1996.
12. Bernhard Schölkopf, Alex Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
13. Matthew Turk and Alexander P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 4(1):71–86, 1991.

文的还阅读了：

义典型相关分析的仿射不变特征提取方法

cal Correlation Analysis for Multiview Semisupervised Feature Extraction

型相关性分析的过程监控系统

型相关分析的组合特征抽取及脸像鉴别

影分析在特征抽取中的应用研究

DA和GSVD的多模态特征提取方法研究

ter space reduction using approximate canonical correlation analysis

义局部判别型典型相关分析算法

sed security assessment of power system using generalized regression neural network with feature extraction

M, METHOD AND COMPUTER PROGRAM PRODUCT FOR DOCUMENT IMAGE ANALYSIS USING

义均值的鲁棒典型相关分析算法

inative Feature Extraction by a Neural Implementation of Canonical Correlation Analysis

eature Extraction Using Principal Component Analysis

嵌入的广义多重集典型相关分析

c feature extraction using discriminant diffusion mapping analysis for automated tool wear evaluation

, method and computer program product for document image analysis using feature extraction functions

型相关与最大散度差的特征抽取方法

Extraction Using Supervised Spectral Analysis

el Feature Extraction Method for Epilepsy EEG Signals Based on Robust Generalized Synchrony Analysis

CTION OF MULTI-AGENT ADVERSARIAL MOVEMENTS THROUGH SIGNATURE-FORMATIONS U

VEL ALGORITHM FOR VOICE CONVERSION USING CANONICAL CORRELATION ANALYSIS

ric feature extraction using local fractal auto-correlation

Extraction of Multimodal Data by Cluster-based Correlation Discriminative Analysis

sed security assessment of power system using generalized regression neural network with feature extraction

Extraction and Analysis on Xinjiang Uygur medicine Image by Using Color Histogram