

Kernel Machine Based Learning  
For Multi-View Face Detection and Pose Estimation

Stan Z. Li, Lie Gu, Bernhard Schölkopf, Hongjiag Zhang  
Contact: szli@microsoft.com

QingDong Fu, Yimin Cheng  
Dept. of Electronic Science and Technology  
University of Science and Technology of China

March 2001  
Technical Report  
MSR-TR-2001-07

Microsoft Research  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052  
<http://www.research.microsoft.com>

## Abstract

Face images are subject to effects of view changes and artifacts such as variations in illumination and facial shape. Such effects cause the distribution of data points to be highly nonlinear and complex. It is desirable to learn a nonlinear mapping from the input image to a low dimensional space such that the distribution becomes simpler, tighter and therefore more predictable for better modeling of faces.

In this paper, we present a kernel machine based approach for learning such nonlinear mappings to provide effective view-based representations for multi-view face detection and pose estimation. One view-subspace is learned for each view from a set of face images of that view, by using kernel principal component analysis (KPCA). Projections of the data onto the view-subspaces are then computed as view-based features. Multi-view face detection and pose estimation are performed by classifying each face into one of the facial views or into the nonface class, by using a multi-class kernel support vector classifier (KSVC). It is shown that fusion of evidences from all views can produce better results than using the result for a single view. Experimental results show that our approach yields high detection and low false alarm rates in face detection and good accuracy in pose estimation, and outperforms its linear counterpart composed of linear principal component analysis (PCA) feature extraction and Fisher linear discriminant based classification (FLDC).

## 1 Introduction

In the past most research on face detection focused on frontal faces (see *e.g.* [21, 10, 14, 16, 20]). However, approximately 75% of the faces in home photos are non-frontal [8] and therefore a system for frontal faces only is very limiting.

Multi-view face detection and pose estimation require to model faces seen from various view points, under variations in illumination and facial shape. Appearance based methods [11, 12, 7] avoid difficulties in 3D modeling by using images or appearances of the object viewed from possible viewpoints. The appearance of an object in a 2D image depends on its shape, reflectance property, pose as seen from the viewing point, and the external illumination conditions. The object is modeled by a collection of appearances parameterized by pose and illumination. Object detection and recognition are performed by comparing the appearances of the object in the image and in the model.

In view-based representation, the view is quantized into a set of discrete values such as the view angles. A view subspace defines the manifold of possible appearances of the object at that view, subject to illumination. One may use one of the following two methods when constructing subspaces: (1) Quantize the pose into several discrete

ranges and partition the data set into several subsets, each composed of data at a particular view; then construct a subspace from each subset [15]. (2) With training data labeled and sorted according to the view value (and perhaps also illumination values), one may be able to construct a manifold describing the distribution across views [11, 5, 1].

Linear principal component analysis (PCA) is a powerful technique for data reduction and feature extraction from high dimensional data. It has been used widely in appearance based applications such as face detection and recognition [21, 10, 20]. The theory of PCA is based on an assumption that the data is a Gaussian distribution. PCA gives accurate density models when the assumption is valid. This is, however, not the case in many real-world applications.

The distribution of face images under a perceivable variation in viewpoint, illumination or expression is highly nonconvex and complex [2], and can hardly be well described by using linear PCA. To obtain a better description of the variations, nonlinear methods, such as principal curves [6] and splines [11], may be used. Over the last years, progress has been made for non-frontal faces. Wiskott *et al.* [24] build elastic bunch graph templates to describe some key facial feature points and their relationships and use them for multi-view face detection and recognition.

Gong and colleagues study the trajectories of faces in linear PCA feature spaces as they rotate [5], and use kernel support vector machines for multi-pose face detection and pose estimation [13, 9]. Their systems use not only information on the face appearance but also constraints from color and motion.

Schneiderman and Kanade [17] use a statistical model to represent object's appearances over a small range of views, to capture variation that cannot be modeled explicitly. This includes variation in the object itself, variation due to lighting, and small variations in pose. Another statistical model is used to describe non-objects-of-interest. Each detector is designed for a specific view of the object, and multiple detectors that span a range of the object's orientation are used. The results of these individual detectors are then combined.

In the present paper, we present a kernel machine learning based approach for extracting nonlinear features of face images and using them for multi-view face detection and pose estimation. Kernel PCA [19] is applied on a set of view-labeled face images to learn nonlinear view-subspaces. Nonlinear features are the projections of the data onto these nonlinear view-subspaces.

KPCA feature extraction effectively acts a nonlinear mapping from the input space to an implicit high dimensional feature space. It is hoped that the distribution of the

mapped data in the implicit feature space has a simple distribution so that a simple classifier (which need not to be a linear one) in the high dimensional space could work well.

Face detection and pose estimation are jointly performed by using kernel support vector classifiers (KSVC's), based on the nonlinear features. The main operation here is to classify a windowed pattern into one of the view classes plus the nonface class. In this multi-class classification task, evidences from different view channels are effectively fused to yield a better result than can be produced by any single channel.

Results show that the proposed approach yields high detection and low false alarm rates in face detection, and good accuracy in pose estimation. These are compared with results obtained by using a linear counterpart of our system, i.e., a system building on linear PCA and linear classification methods.

The remainder of the paper is organized as follows: Section 2 introduces basic concepts of kernel learning methods, that is, KSVC and KPCA. Section 3 described the proposed approach for face detection and pose estimation and presents a system implementing our methods. Section 4 presents experimental results.

## 2 Kernel Learning Methods

The kernel methods generalize linear SVC and PCA to nonlinear ones. The trick of kernel methods is to perform dot products in the feature space by using kernel functions in input space so that the nonlinear mapping is performed implicitly in the input space [22, 19].

### 2.1 Support Vector Classifier

Consider the problem of separating the set of training vectors belonging to two classes, given a set of training data  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$  where  $\mathbf{x}_i \in \mathbb{R}^N$  is a feature vector and  $y_i \in \{-1, +1\}$  its class label.

Assume (1) that the two classes can be separated by a hyperplane  $\mathbf{w} \cdot \mathbf{x} + b = 0$  and (2) no knowledge about the data distribution is available. From the point of view of statistical learning theory, of all the boundaries determined by  $\mathbf{w}$  and  $b$ , the one that maximizes the margin is preferable, due to a bound on its expected generalization error. The optimal values for

$\mathbf{w}$  and  $b$  can be found by solving the following constrained minimization problem

$$\min_{\mathbf{w}} E(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 \quad (1)$$

$$\text{s.t. } y_i [(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 \quad i = 1, \dots, m \quad (2)$$

Solving it requires the construction of a so-called dual problem, using Lagrange multipliers  $\alpha_i$  ( $i = 1, \dots, m$ ),

and results in a classification function

$$f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^m \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \right) \quad (3)$$

Most of the  $\alpha_i$  take the value of zero; those  $\mathbf{x}_i$  with nonzero  $\alpha_i$  are the ‘‘support vectors’’.

In non-separable cases, slack variables  $\xi_i \geq 0$  measuring the misclassification errors can be introduced, and a penalty function added to the objective function [4]. The optimization problem is now treated as to minimize an upper bound on the total classification error as well as, approximately, a bound on the VC dimension of the classifier. The solution is identical to the separable case except for a modification of the Lagrange multipliers into  $0 \leq \alpha_i \leq C$ ,  $i = 1, \dots, m$  [22].

A linearly non-separable but nonlinearly (better) separable case may be tackled as follows: First, map the data from the input space  $\mathbb{R}^N$  to a high dimensional feature space  $\mathbb{F}$  by  $\mathbf{x} \rightarrow \Phi(\mathbf{x}) \in \mathbb{F}$  such that the mapped data is linearly separable in the feature space. Assuming there exists a kernel function  $K$  such that  $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ , then a nonlinear SVM can be constructed by replacing the inner product  $\mathbf{x} \cdot \mathbf{y}$  in the linear SVM by the kernel function  $K(\mathbf{x}, \mathbf{y})$

$$f(\mathbf{x}) = \text{sign} \left( \sum_{i=1}^m \bar{\alpha}_i y_i K(\mathbf{x}_i \cdot \mathbf{x}) + \bar{b} \right) \quad (4)$$

This corresponds to constructing an optimal separating hyperplane in the feature space.

In solving the  $L + 1$  class problem, the one-against-the-rest method [18, 3, 23] is used to construct  $L + 1$  classifiers. The  $k$ -th one separates class  $k$  from the other  $L$  classes. A maximum selection across the classifiers or some another measure is used for the final decision.

### 2.2 Kernel PCA

We begin by describing linear PCA. Given a set of examples in  $\mathbb{R}^N$  represented by column vectors, subtract them by their mean vector to obtain the centered examples  $\mathbf{x}_i \in \mathbb{R}^N$  ( $i = 1, \dots, m$ ). The covariance matrix is

$$\mathbf{C} = \frac{1}{m} \sum_{j=1}^m \mathbf{x}_j \mathbf{x}_j^T. \quad (5)$$

Linear PCA is an algorithm which diagonalizes the covariance matrix by performing a linear transformation. The corresponding transformation matrix is constructed by solving the following eigenvalue problem

$$\lambda \mathbf{v} = \mathbf{C} \mathbf{v} \quad (6)$$

for eigenvalues  $\lambda_i \geq 0$  and nonzero eigenvectors. The above is equivalent to

$$\lambda(\mathbf{x}_k \cdot \mathbf{v}) = (\mathbf{x}_k \cdot \mathbf{C} \mathbf{v}) \quad \forall k = 1, \dots, m \quad (7)$$

Sort  $\lambda_i$  in descending order and use the first  $M \leq N$  principal components  $\mathbf{v}_i$  as the basis vector of a lower dimensional subspace. The transformation matrix can be formed by using  $\mathbf{v}_i$ , normalized to unit length, as the  $i$ -th column of a matrix  $\mathbf{T}$ . The projection of a point  $\mathbf{x} \in \mathbb{R}^N$  into the  $M$ -dimensional subspace can be calculated as

$$\alpha = (\alpha_1, \dots, \alpha_M) = \mathbf{x}^\top \mathbf{T} \in \mathbb{R}^M \quad (8)$$

Its reconstruction from  $\alpha$  is

$$\hat{\mathbf{x}} = \sum_{i=1}^M \alpha_i \mathbf{v}_i. \quad (9)$$

This is the best approximation of the  $\mathbf{x}_1, \dots, \mathbf{x}_m$  in any  $M$ -dimensional subspace in the sense of minimum overall squared error.

Let us now generalize classic PCA to kernel PCA. Let  $\phi : \mathbf{x} \in \mathbb{R}^N \rightarrow \mathbf{X} \in \mathbb{H}$  be a mapping from the input space to a high dimensional feature space. Now we introduce the idea of KPCA by which a nonlinear PCA representation in the input space is obtained by using a linear PCA in  $\mathbb{H}$ . The covariance matrix in  $\mathbb{H}$  is

$$\overline{\mathbf{C}} = \frac{1}{m} \sum_{j=1}^m \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_j)^T \quad (10)$$

and the eigenvalue problem is  $\lambda \mathbf{V} = \overline{\mathbf{C}} \mathbf{V}$ . Corresponding to Eq.(7) are the equations in  $\mathbb{H}$

$$\lambda (\Phi(\mathbf{x}_k) \cdot \mathbf{V}) = (\Phi(\mathbf{x}_k) \cdot \mathbf{C} \mathbf{V}) \quad \forall k = 1, \dots, m \quad (11)$$

Because all  $\mathbf{V}$  for nonzero  $\lambda$  must lie in the span of the  $\mathbf{x}_k$ 's, there exist coefficients  $\alpha_i$  such that

$$\mathbf{V} = \sum_{i=1}^m \alpha_i \Phi(\mathbf{x}_i) \quad (12)$$

Defining the matrix  $K = [K_{i,j}]_{m \times m}$ , the eigenvalue problem can be converted into the following [19]

$$m \lambda \alpha = K \alpha \quad (13)$$

for nonzero eigenvalues.

Sort  $\lambda_i$  in descending order and use the first  $M \leq m$  principal components  $\mathbf{V}_i$  as the basis vector in  $\mathbb{H}$  (In fact, there are usually some zero eigen-values, in which case  $M < m$ ). The  $M$  vectors spans a linear subspace, called KPCA subspace, in  $\mathbb{H}$ . The projection of a point  $\mathbf{x}$  onto the  $k$ -th kernel principal component  $\mathbb{V}_k$  is calculated as

$$\begin{aligned} \beta_k = (\mathbf{V}_k \cdot \Phi(\mathbf{x})) &= \sum_{i=1}^m \alpha_{k,i} (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) \\ &= \sum_{i=1}^m \alpha_{k,i} K(\mathbf{x}_i, \mathbf{x}) \end{aligned} \quad (14)$$

### 3 Multi-View Face Detection and Pose Estimation

In the following, the problem of multi-view face detection and pose estimation is formulated. The proposed kernel approach for solving the problem is described.

#### 3.1 Problem Description

Let  $\mathbf{x} \in \mathbb{R}^N$  be a windowed grey-level image, or appearance, of a face, possibly preprocessed. Assume that all left rotated faces (those with view angles between  $91^\circ$  and  $180^\circ$ ) are mirrored to right rotated so that every view angle is between  $0^\circ$  and  $90^\circ$ ; this does not cause any loss of generality. Quantize the pose into a set of  $L$  discrete values. We choose  $L = 10$  for 10 equally spaced angles  $\theta_0 = 0^\circ$ ,  $\theta_1 = 10^\circ, \dots, \theta_9 = 90^\circ$ , with  $0^\circ$  corresponding to the right side view and  $90^\circ$  to the frontal view.

Assume that a set of training face images are provided for the learning; see Fig. 1 for some examples. The images  $\mathbf{x}$  are subject to changes not only in the view  $\theta$ , but also in illumination. The training set is view-labeled in that each face image is manually labeled with its view value as close to the truth as possible, and then assigned into one of  $L$  groups according to the nearest view value. This produces  $L$  view-labeled face image subsets for learning view-subspaces of faces. Another training set of nonface images is also needed for training face detection.

Now, there are  $L + 1$  classes. This will be indexed in the following by  $\ell$ , with  $\ell \in \{0, 1, \dots, L - 1\}$  corresponding to the  $L$  views of faces and  $\ell = L$  corresponding to the nonface class. The two tasks, face detection and pose estimation, are performed jointly by classifying the input  $\mathbf{x}$  into one of the  $L + 1$  classes. If the input is classified into one of the  $L$  face classes, a face is detected and the corresponding view is the estimated pose; otherwise the input pattern is considered as a nonface pattern (without a view).

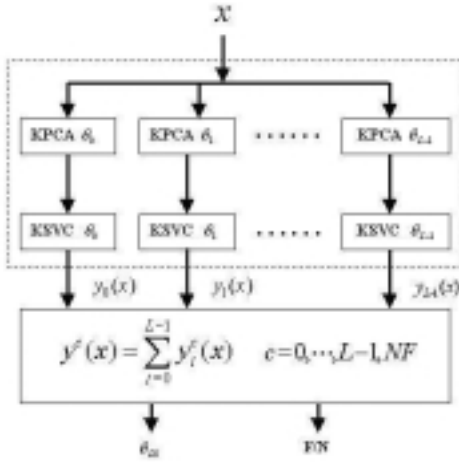
#### 3.2 Kernel Machine Based Learning

The learning for face detection and pose estimation using kernel machines is carried out in two stage: one for KPCA view-subspace learning, and one for KSVC classifier training. This is illustrated through the system structure shown in Fig. 2.

Stage 1 training aims to learn the  $L$  KPCA view-subspaces from the  $L$  face view subsets. One set of ker-



Figure 1. Multi-view face examples.



**Figure 2. The structure of the multiple KPCA and SVCs and the composite face detector and pose estimator.**

nel principal components (KPCs)  $\alpha$  are learned from each view subset. The first  $M = 50$  most significant components are used as the basic vectors to construct the view-subspace. The learning in this stage yields  $L$  view-subspaces, each determined by a set of support vectors and the corresponding coefficients. The KPCA in each view channel effectively performs a nonlinear mapping from the input image space (possibly pre-processed) to the  $M = 50$  dimensional output KPCA feature space.

Stage 2 aims to train  $L$  KSVC's to differentiate between face and nonface patterns for face detection. This requires a training set consisting of a nonface subset as well as  $L$  view face subsets, as mentioned earlier. One KSVC is trained for each view to perform the  $L + 1$ -class classification based on the features in the corresponding KPCA subspace. The projection onto the KPCA subspace of the corresponding view is used as the feature vector. The one-against-the-rest method [18, 3, 23] is used for solving the multi-class problem in a KSVC. This stage gives  $L$  KSVCs.

### 3.3 Face Detection and Pose Estimation

In the testing stage, a test sample is presented to the KPCA feature extractor for each view  $\ell$  to obtain the feature vector for that view. The corresponding KSVC of that view calculates an output vector  $\mathbf{y}_\ell = (y_\ell^c \mid c = 0, \dots, L)$  as the responses of the  $L + 1$  classes to the input. This is done for all the  $L$  view channels so that  $L$  such output vectors  $\{\mathbf{y}_\ell \mid \ell = 0, \dots, L - 1\}$  are produced.

The value  $y_\ell^c$  is the evidence for the judgement that the input  $\mathbf{x}$  belongs to class  $c$  in terms of the features in the  $\ell$ -th view KPCA subspace. The final classification decision is

made by fusing the evidences from all the  $L$  view channels. A simple way for the fusing is to sum the evidences; that is, for each class  $c = 0, \dots, L$ , the following

$$y^c(\mathbf{x}) = \sum_{\ell=0}^{L-1} y_\ell^c \quad (15)$$

is calculated to give the overall evidence for classifying  $\mathbf{x}$  into class  $c$ . The final decision is made by maximizing evidence:  $\mathbf{x}$  belongs to  $c^*$  if  $c^* = \arg \max_c y^c(\mathbf{x})$ .

## 4 Experimental Result

### 4.1 Data Description

A data set consisting of  $L$  face view subsets and a non-face subset is given. It is randomly partitioned into three data sets for the use in different stages. Fig. 1 shows the partition and the sizes of the three data sets. Set 1 is used for learning the  $L$  KPCA view-subspaces, Sets 1 and 2 together are used for training the  $L$  multi-class KSVC's, and Set 3 is used for testing.

**Table 1. Composition of three data sets**

| View      | Set 1 | Set 2 | Set 3 |
|-----------|-------|-------|-------|
| 90°       | 500   | 2000  | 2209  |
| 80°       | 500   | 2000  | 1709  |
| 70°       | 500   | 2000  | 1394  |
| 60°       | 500   | 2000  | 1137  |
| 50°       | 500   | 2000  | 1189  |
| 40°       | 500   | 2000  | 1143  |
| 30°       | 500   | 2000  | 1304  |
| 20°       | 500   | 2000  | 1627  |
| 10°       | 500   | 2000  | 1553  |
| 0°        | 500   | 2000  | 1309  |
| Tot.Faces | 5000  | 20000 | 14574 |
| Nonfaces  | 0     | 10000 | 7849  |

### 4.2 Training

For the KPCA, a polynomial kernel is selected,  $K(\mathbf{x}, \mathbf{y}) = (a(\mathbf{x} \cdot \mathbf{y}) + b)^n$ , with  $a = 0.001$ ,  $b = -1$ ,  $n = 3$ . For the KSVC, an RBF kernel is selected,  $K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x} - \mathbf{y}\|^2 / \sigma^2}$  with  $\sigma = 0.1$ . The selections are empirical.

The quality of the KPCA subspace modeling depends on the size of training data. A larger training data normally leads to a better generalization quality, but also increases the computational costs in both learning and projection, KPCA learning and projection being where most computational expenses occur in our system. Table 2 shows the error rates for various sample size. We use 500 examples per view for the KPCA learning of the view-subspaces to balance the tradeoff.

**Table 2. Error rates with different numbers of training examples per view**

| Using Num. (%) | Missing (%) | False A. |
|----------------|-------------|----------|
| 300            | 8.16        | 0.30     |
| 500            | 6.82        | 0.31     |
| 2000           | 5.13        | 0.21     |
| 3000           | 5.03        | 0.12     |

### 4.3 Test Results

Four methods are compared between the linear PCA based and the KPCA based approaches. The KSVC is used for multi-class classification based on KPCA features, and a linear classifier, *i.e.* Fisher linear discriminant based classifier (FLDC), is used for classification based on linear PCA. The reason for the use of the linear FLD is that a linear SVC would have nearly half of the training samples as its support vectors. The FLD classifier is combined with the one-against-all strategy for the multiple class problem.

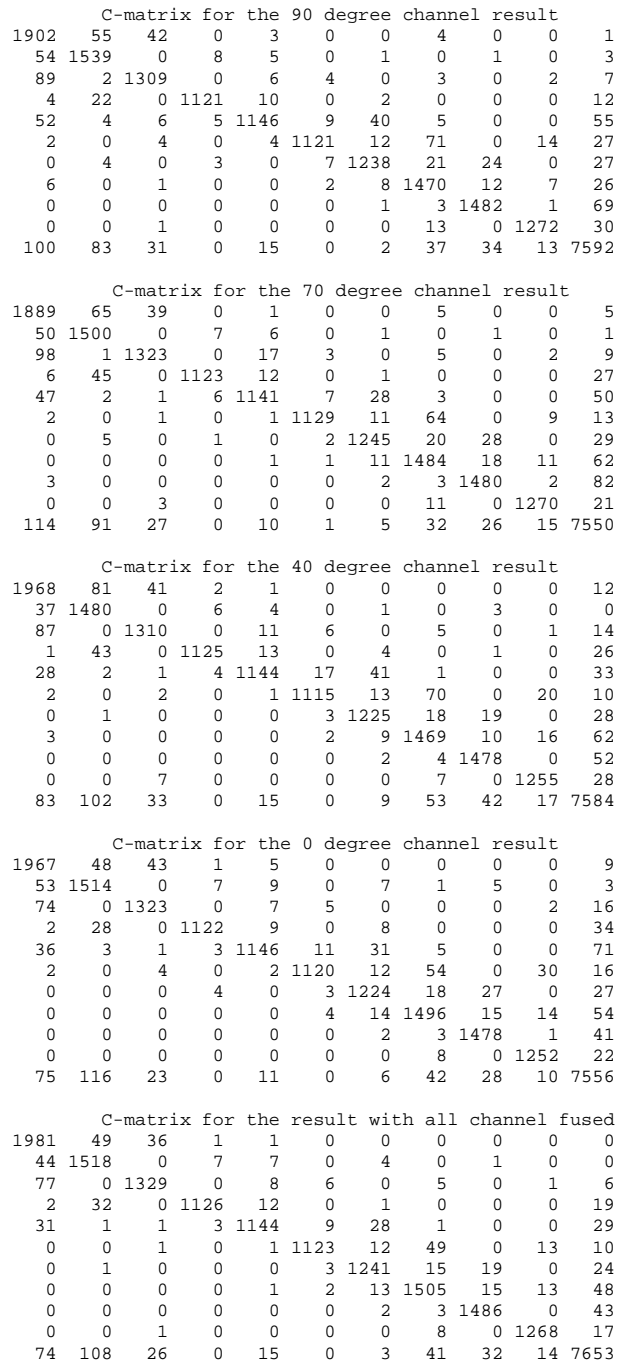
The classification results are demonstrated through classification matrices (c-matrices); see Fig. 3 and 4. The entry  $(i, j)$  of the c-matrix gives the number of examples whose ground truth (manually labeled, actually) class label is  $i$  (in row) and which are classified into class  $j$  (in column) by the system. The first  $L = 10$  rows and 10 columns correspond to the 10 views ( $0^\circ, 10^\circ, \dots, 90^\circ$ ) of the ground truth and the classification result, respectively, whereas the last row and column correspond to the nonface class.

From these c-matrices, the corresponding missing and false alarm rates for face detection can be calculated, as shown in Table 3; and also the accuracy for pose estimation shown in Table 4 where the  $\pm 10^\circ$  accuracy is defined as the percentage of examples whose pose estimates are within the range of  $\pm 10^\circ$  (*i.e.* the elements on the diagonal line and one off-diagonal line on each side) and the  $\pm 20^\circ$  accuracy defined as the percentage of examples whose pose estimates are within the range of  $\pm 20^\circ$  (*i.e.* the elements on the diagonal line and those on the two off-diagonal line on each side). From these we can see the the kernel PCA approach produces much better results than linear PCA.

Finally, the face detection and pose estimation are performed on real images. Testing images collected from VCD movies are used for the evaluation. The images are scanned at different scales and locations. The pattern in each sub-window is classified into face/nonface; if it is a face, the pose is estimated. Fig. 5 shows some examples.

## 5 Conclusion

A kernel machine based approach has been presented for learning view-based representations for multi-view face



**Figure 3. Classification statistics for the kernel PCA approach as demonstrated by c-matrices (see text for the interpretation of c-matrices).**

C-matrix for the 90 degree channel result

|      |     |     |     |     |     |     |     |     |     |      |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1352 | 271 | 37  | 12  | 10  | 3   | 11  | 15  | 68  | 26  | 1382 |
| 241  | 746 | 429 | 234 | 27  | 11  | 4   | 23  | 20  | 42  | 830  |
| 5    | 116 | 284 | 189 | 108 | 27  | 2   | 7   | 12  | 13  | 177  |
| 0    | 20  | 146 | 209 | 190 | 76  | 16  | 25  | 24  | 48  | 256  |
| 21   | 4   | 43  | 119 | 151 | 52  | 59  | 17  | 34  | 8   | 154  |
| 9    | 5   | 55  | 113 | 336 | 394 | 260 | 150 | 206 | 29  | 125  |
| 28   | 10  | 8   | 17  | 152 | 344 | 505 | 442 | 207 | 61  | 178  |
| 15   | 10  | 0   | 0   | 7   | 65  | 170 | 288 | 190 | 69  | 89   |
| 2    | 0   | 0   | 6   | 9   | 3   | 30  | 169 | 218 | 129 | 108  |
| 0    | 19  | 18  | 29  | 25  | 8   | 43  | 192 | 287 | 391 | 186  |
| 536  | 508 | 374 | 209 | 174 | 160 | 204 | 299 | 287 | 493 | 4364 |

C-matrix for the 70 degree channel result

|      |     |     |     |     |     |     |     |     |     |      |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1517 | 422 | 49  | 35  | 11  | 9   | 8   | 31  | 57  | 25  | 1545 |
| 137  | 669 | 352 | 149 | 27  | 6   | 3   | 18  | 23  | 22  | 664  |
| 6    | 121 | 209 | 129 | 54  | 6   | 18  | 17  | 12  | 25  | 172  |
| 0    | 21  | 231 | 304 | 210 | 92  | 6   | 36  | 26  | 54  | 265  |
| 5    | 3   | 34  | 142 | 265 | 191 | 111 | 41  | 56  | 6   | 122  |
| 2    | 0   | 15  | 45  | 240 | 258 | 207 | 125 | 132 | 23  | 66   |
| 5    | 3   | 10  | 5   | 72  | 250 | 285 | 241 | 212 | 27  | 157  |
| 9    | 2   | 2   | 0   | 48  | 153 | 398 | 597 | 340 | 140 | 171  |
| 0    | 2   | 0   | 0   | 0   | 0   | 30  | 78  | 163 | 82  | 76   |
| 0    | 18  | 13  | 21  | 29  | 23  | 24  | 136 | 215 | 299 | 100  |
| 528  | 448 | 479 | 307 | 233 | 155 | 214 | 307 | 317 | 606 | 4511 |

C-matrix for the 40 degree channel result

|      |     |     |     |     |     |     |     |     |     |      |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1484 | 355 | 50  | 11  | 4   | 9   | 3   | 20  | 24  | 7   | 1302 |
| 349  | 741 | 344 | 124 | 19  | 3   | 2   | 8   | 13  | 26  | 1128 |
| 14   | 234 | 489 | 361 | 117 | 24  | 12  | 10  | 23  | 18  | 234  |
| 0    | 14  | 81  | 182 | 138 | 61  | 6   | 11  | 20  | 26  | 132  |
| 0    | 0   | 61  | 185 | 338 | 216 | 110 | 37  | 63  | 0   | 108  |
| 0    | 0   | 3   | 46  | 244 | 288 | 259 | 163 | 164 | 22  | 79   |
| 22   | 2   | 4   | 5   | 58  | 250 | 458 | 510 | 269 | 81  | 287  |
| 2    | 0   | 3   | 0   | 37  | 56  | 196 | 225 | 199 | 52  | 97   |
| 11   | 2   | 4   | 3   | 8   | 9   | 40  | 190 | 208 | 130 | 129  |
| 3    | 17  | 20  | 29  | 26  | 12  | 33  | 139 | 222 | 292 | 150  |
| 324  | 344 | 335 | 191 | 200 | 215 | 185 | 314 | 348 | 655 | 4203 |

C-matrix for the 0 degree channel result

|      |      |     |     |     |     |     |     |     |     |      |
|------|------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 47   | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 15   |
| 1539 | 1174 | 363 | 91  | 5   | 0   | 0   | 2   | 12  | 30  | 1682 |
| 29   | 249  | 320 | 206 | 40  | 12  | 0   | 0   | 3   | 3   | 128  |
| 0    | 22   | 106 | 75  | 48  | 17  | 0   | 1   | 1   | 13  | 52   |
| 92   | 89   | 286 | 442 | 370 | 221 | 81  | 46  | 64  | 14  | 815  |
| 130  | 5    | 17  | 79  | 258 | 346 | 275 | 195 | 210 | 58  | 420  |
| 114  | 12   | 0   | 4   | 40  | 192 | 465 | 511 | 213 | 87  | 504  |
| 53   | 10   | 7   | 0   | 5   | 3   | 48  | 89  | 108 | 19  | 143  |
| 60   | 36   | 13  | 3   | 7   | 1   | 51  | 291 | 371 | 291 | 466  |
| 0    | 0    | 11  | 9   | 0   | 0   | 4   | 13  | 48  | 107 | 13   |
| 145  | 112  | 271 | 228 | 416 | 351 | 380 | 479 | 523 | 687 | 3611 |

C-matrix for the result with all channel fused

|      |     |     |     |     |     |     |     |     |     |      |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|
| 1473 | 284 | 27  | 5   | 2   | 2   | 1   | 9   | 30  | 8   | 1265 |
| 321  | 889 | 422 | 158 | 23  | 2   | 1   | 6   | 16  | 13  | 1180 |
| 14   | 151 | 319 | 223 | 95  | 8   | 12  | 9   | 7   | 14  | 206  |
| 0    | 25  | 192 | 256 | 167 | 86  | 8   | 26  | 24  | 28  | 207  |
| 15   | 3   | 64  | 218 | 357 | 239 | 122 | 48  | 65  | 4   | 212  |
| 13   | 0   | 12  | 31  | 213 | 260 | 222 | 108 | 163 | 33  | 119  |
| 19   | 12  | 4   | 4   | 93  | 308 | 481 | 457 | 262 | 73  | 267  |
| 4    | 0   | 0   | 0   | 18  | 55  | 200 | 341 | 212 | 62  | 113  |
| 16   | 8   | 0   | 0   | 9   | 6   | 50  | 255 | 306 | 198 | 205  |
| 0    | 17  | 18  | 33  | 25  | 0   | 13  | 109 | 200 | 331 | 157  |
| 334  | 320 | 336 | 209 | 187 | 177 | 194 | 259 | 268 | 545 | 3918 |

**Figure 4. Classification statistics for the linear PCA approach as demonstrated by c-matrices.**

**Table 3. Face Detection Error Rates**

| Method                | Missing (%) | False A. (%) |
|-----------------------|-------------|--------------|
| KPCA-90° + KSVC       | 2.16        | 3.27         |
| KPCA-70° + KSVC       | 2.20        | 3.81         |
| KPCA-40° + KSVC       | 2.43        | 3.38         |
| KPCA-0° + KSVC        | 2.13        | 3.73         |
| KPCA + L KSVC's fused | 2.15        | 2.50         |
| PCA-90° + FLD         | 22.26       | 44.40        |
| PCA-70° + FLD         | 24.66       | 42.53        |
| PCA-40° + FLD         | 21.34       | 46.55        |
| PCA-0° + FLD          | 24.65       | 54.00        |
| PCA + L FLDC's fused  | 19.41       | 50.18        |

**Table 4. Face Pose Estimation Accuracy**

| Method                | ±20° Acc. | ±10° Acc. |
|-----------------------|-----------|-----------|
| KPCA-90° + KSVC       | 99.14     | 96.80     |
| KPCA-70° + KSVC       | 99.27     | 96.84     |
| KPCA-40° + KSVC       | 99.35     | 97.0      |
| KPCA-0° + KSVC        | 99.11     | 97.06     |
| KPCA + L KSVC's fused | 99.46     | 97.52     |
| PCA-90° + FLD         | 92.10     | 81.40     |
| PCA-70° + FLD         | 93.15     | 83.34     |
| PCA-40° + FLD         | 93.68     | 84.25     |
| PCA-0° + FLD          | 91.47     | 83.38     |
| PCA + L FLDC's fused  | 94.06     | 84.56     |

detection and recognition. The main part of the work is the use of KPCA for extracting nonlinear features for each view by learning the nonlinear view-subspace using kernel PCA. This is to construct a mapping from the input image space, in which the distribution of data points is highly nonlinear and complex, to a lower dimensional space in which the distribution becomes simpler, tighter and therefore more predictable for better modeling of faces.

The kernel learning approach leads to an architecture composed of an array of KPCA feature extractors, one for each view, and an array of corresponding KSVC multi-class classifiers for face detection and pose estimation. Evidences from all views are fused to produce better results than the result from a single view. Results show that the kernel learning approach outperforms its linear counterpart and yields high detection and low false alarm rates in face detection, and good accuracy in pose estimation.



**Figure 5. Multi-view face detection results. The estimated views are as follows: From left to right, in the two images on the top, the estimated angles are 10, 0, 60, 50 degrees, respectively; in the bottom image, they are 80, 60, 0 degrees.**

## References

- [1] S. Baker, S. Nayar, and H. Murase. Parametric feature detection. *IJCV*, 27(1):27–50, March 1998.
- [2] M. Bichsel and A. P. Pentland. “Human face recognition and the face image set’s topology”. *CVGIP: Image Understanding*, 59:254–261, 1994.
- [3] V. Blanz, B. Schölkopf, H. Bülthoff, C. Burges, V. Vapnik, and T. Vetter. Comparison of view-based object recognition algorithms using realistic 3D models. In C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff, editors, *Artificial Neural Networks — ICANN’96*, pages 251 – 256, Berlin, 1996. Springer Lecture Notes in Computer Science, Vol. 1112.
- [4] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.
- [5] S. Gong, S. McKenna, and J. Collins. “An investigation into face pose distribution”. In *Proc. IEEE International Conference on Face and Gesture Recognition*, Vermont, 1996.
- [6] T. Hastie and W. Stuetzle. “Principal curves”. *Journal of the American Statistical Association*, 84(406):502–516, 1989.
- [7] J. Hornegger, H. Niemann, and R. Risack. Appearance-based object recognition using optimal feature transforms. *PR*, 33(2):209–224, February 2000.
- [8] A. Kuchinsky, C. Pering, M. L. Creech, D. Freeze, B. Serra, and J. Gwizdka. “FotoFile: A consumer multimedia organization and retrieval system”. In *Proc. ACM HCI’99 Conference*, 1999.
- [9] Y. M. Li, S. G. Gong, and H. Liddell. “support vector regression and classification based multi-view face detection and recognition”. In *IEEE Int. Conf. On Face & Gesture Recognition*, pages 300–305, France, 2000.
- [10] B. Moghaddam and A. Pentland. “Probabilistic visual learning for object representation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:696–710, July 1997.
- [11] H. Murase and S. K. Nayar. “Visual learning and recognition of 3-D objects from appearance”. *International Journal of Computer Vision*, 14:5–24, 1995.
- [12] S. Nayar, S. Nene, and H. Murase. Subspace methods for robot vision. *RA*, 12(5):750–758, October 1996.
- [13] J. Ng and S. Gong. “performing multi-view face detection and pose estimation using a composite support vector machine across the view sphere”. In *Proc. IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, pages 14–21, Corfu, Greece, September 1999.
- [14] E. Osuna, R. Freund, and F. Girosi. “Training support vector machines: An application to face detection”. In *CVPR*, pages 130–136, 1997.
- [15] A. P. Pentland, B. Moghaddam, and T. Starner. “View-based and modular eigenspaces for face recognition”. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 84–91, 1994.
- [16] H. A. Rowley, S. Baluja, and T. Kanade. “Neural network-based face detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–28, 1998.
- [17] H. Schneiderman and T. Kanade. “a statistical method for 3d object detection applied to faces and cars. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000.

- [18] B. Schölkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings, First International Conference on Knowledge Discovery & Data Mining*, Menlo Park, 1995. AAAI Press.
- [19] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998. Technical Report No. 44, 1996, Max Planck Institut für biologische Kybernetik, Tübingen.
- [20] K.-K. Sung and T. Poggio. “Example-based learning for view-based human face detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.
- [21] M. A. Turk and A. P. Pentland. “Face recognition using eigenfaces.”. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, Hawaii, June 1991.
- [22] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.
- [23] J. Weston and C. Watkins. Multi-class support vector machine. Technical Report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998.
- [24] L. Wiskott, J. Fellous, N. Kruger, and C. V. malsburg. “face recognition by elastic bunch graph matching”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.