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

Stan Z. Li[†], QingDong Fu[†‡], Lie Gu[†], Bernhard Schölkopf[†], Yimin Cheng[‡], HongJiag Zhang[†]
[†] Microsoft Research China, Beijing Sigma Center, Beijing 100080, China
[‡] Dept. of Electronic Science and Technology, University of Science and Technology of China
Contact: szli@microsoft.com, http://research.microsoft.com/~szli

Abstract

Face images are subject to changes in view and illumination. Such changes cause data distribution to be highly nonlinear and complex in the image space. It is desirable to learn a nonlinear mapping from the image space 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. The aim is to provide an effective view-based representation for multi-view face detection and pose estimation. Assuming that the view is partitioned into a number of distinct ranges, one nonlinear view-subspace is learned for each (range of) view from a set of example face images of that view (range), by using kernel principal component analysis (KPCA). Projections of the data onto the view-subspaces are then computed as view-based nonlinear features. Multi-view face detection and pose estimation are performed by classifying a face into one of the facial views or into the nonface class, by using a multi-class kernel support vector classifier (KSVC). Experimental results show that fusion of evidences from multi-views can produce better results than using the result from a single view; and that our approach yields high detection and low false alarm rates in face detection and good accuracy in pose estimation, in comparison with the linear counterpart composed of linear principal component analysis (PCA) feature extraction and Fisher linear discriminant based classification (FLDC).

1 Introduction

Statistics show that approximately 75% of the faces in home photos are non-frontal [21] and it is important to deal with multi-view faces in many face-related applications. While in the past, research on face detection is focused on frontal faces (see *e.g.* [37, 23, 27, 29, 35]), recent years has seen progress in non-frontal faces detection and recognition. Feraud *et al.* [14] and Schneiderman and Kanade [30] adopt the view-based representation [28] in face detection. Wiskott *et al.* [40] build elastic bunch graph templates for multi-view face detection and recognition. Gong and colleagues study the trajectories of faces in linear PCA feature spaces as they rotate [16], and use kernel support vector machines (SVMs) for multi-pose face detection and pose estimation [26, 22]. Huang *et al.* [19] use an SVM to classify between three face poses viewed at $-33.75, 0, +33.75$ degrees.

Linear principal component analysis (PCA), as a technique for data reduction and feature extraction, has been used widely in appearance based applications such as face detection and recognition [37, 23, 35]. However, the distribution of face images under changes in viewpoint and illumination is nonconvex and too complex to be well described by using a single linear PCA [6]. Multi-view face detection and pose estimation require to model faces seen from various view points, under variations in illumination and facial shape. The appearance based vision (see *e.g.* [20, 37, 5, 28, 24, 25, 18]) attempts to avoid difficulties encountered in traditional 3-D vision by modeling a 3-D object using its 2-D appearances. Much research has been done to derive a representation which takes into account changes in viewpoint and illumination [9, 17, 12, 1, 3, 13, 33, 2, 4, 15, 10, 41, 18, 34]. In view-based representation, a view subspace is defined to model the manifold of possible appearances of the object at that view [28]. With training data labeled the view value (and possibly illumination values), one may be able to construct a parametric distribution describing the distribution across views [24, 16, 2]. However, a single linear model can hardly provide a solution for the problem. A mixture of probabilistic PCA subspaces [36, 7], a generative model, may be used for this.

In this paper, we present a kernel machine learning based approach for extracting nonlinear features of face images for multi-view face detection and pose estimation. Kernel

0-7695-1143-0/01 $10.00 © 2001 IEEE

PCA [32] is applied to a set of view-labeled face images to learn nonlinear view-subspaces (Section 2). Nonlinear features are the projections of the data onto these nonlinear view-subspaces. The 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 feature space has a simple distribution so that a simple classifier could do a proper job.

After the KPCA feature extraction, face detection and pose estimation are jointly performed by using kernel support vector classifiers (KSVC's) (Section 3). The aim 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 (Section 4), in comparison with a linear counterpart on linear PCA and linear classification methods.

2 Kernel Learning Methods

The kernel methods generalize linear SVC and linear PCA to nonlinear ones using the "kernel trick" [38, 32]: An implicitly nonlinear mapping in the input space is performed by performing dot products in the feature space, by using kernel functions.

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 $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ where $\mathbf{x}_i \in R^N$ 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. Then, of all the boundaries determined by $\mathbf{w}$ and b, the optimal one having the lowest bound on the expected generalization error is that maximizes the margin [38]. The optimal values for $\mathbf{w}$ and b can be found by solving the following constrained minimization problem

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

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

Solving it using Lagrange multipliers α_i $(i = 1, \ldots, m)$ results in a classification function

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

where α_i and b are found by using an SVC learning algorithm [11, 38]. Most of the α_i's take the value of zero; those $\mathbf{x}_i$ with nonzero α_i are the "support vectors". In cases where the two classes are non-separable [11, 38], the solution is identical to the separable case except for a modification of the Lagrange multipliers into $0 \leq \alpha_i \leq C$, $i = 1, \ldots, m$ where C is the penalty for a mis-classification.

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 $\mathbf{x} \to \Phi(\mathbf{x}) \in \mathbb{F}$ such that the mapped data points of the two classes are 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 $K(\mathbf{x}, \mathbf{y})$

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

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

2.2 Kernel PCA

Given a set of examples in $\mathbb{R}^N$ represented by column vectors, subtract them by the their mean vector to obtain the centered examples $\mathbf{x}_i \in \mathbb{R}^N$ $(i = 1, \ldots, m)$. The covariance matrix can be computed as $\mathbf{C} = \frac{1}{m} \sum_{j=1}^{m} \mathbf{x}_j \mathbf{x}_j^T$. Linear PCA diagonalizes the covariance matrix by performing a linear transformation of which the matrix is constructed by solving the eigenvalue problem $\lambda \mathbf{v} = \mathbf{C}\mathbf{v}$, *i.e.*

$$\lambda(\mathbf{x}_i \cdot \mathbf{v}) = (\mathbf{x}_i \cdot \mathbf{C}\mathbf{v}) \quad \forall i = 1, \ldots, m \qquad (5)$$

Sort the eigenvalues in descending order, and use the first $M \leq N$ principal components $\mathbf{v}_k$ $(1 \leq k \leq M)$ as the basis vector of a lower dimensional subspace, forming the transformation matrix $\mathbf{T}$. The projection of a point $\mathbf{x} \in \mathbb{R}^N$ into the M-dimensional subspace can be calculated as $\alpha = (\alpha_1, \ldots, \alpha_M) = \mathbf{x}^\top \mathbf{T} \in \mathbb{R}^M$. Its reconstruction from α is $\hat{\mathbf{x}} = \sum_{i=1}^{M} \alpha_i \mathbf{v}_i$. This is the best approximation of the $\mathbf{x}_1, \ldots, \mathbf{x}_m$ in any M-dimensional subspace in the sense of minimum squared error.

Kernel PCA is performed by mapping the data from the input space to a high dimensional feature space using $\Phi : \mathbf{x} \in \mathbb{R}^N \to \mathbf{X} \in \mathbb{F}$, and then perform a linear PCA in $\mathbb{F}$. The covariance matrix in $\mathbb{F}$ is $\overline{\mathbf{C}} = \frac{1}{m} \sum_{j=1}^{m} \Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_j)^T$, and the eigenvalue problem is $\lambda \mathbf{V} = \overline{\mathbf{C}}\mathbf{V}$. Corresponding to Eq.(5) are the equations in $\mathbb{F}$

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{V}) = (\Phi(\mathbf{x}_k) \cdot \mathbf{C}\mathbf{V}) \quad \forall k = 1, \ldots, m \qquad (6)$$

Because all $\mathbf{V}$ for nonzero λ must lie in the span of the $\Phi(\mathbf{x}_i)$'s, there exist coefficients α_i such that

$$\mathbf{V} = \sum_{i=1}^{m} \alpha_i \Phi(\mathbf{x}_i) \qquad (7)$$

Defining the matrix $K = [K_{i,j}]_{m\times m}$, where $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$, the eigenvalue problem can be converted into the following [32]

$$m\lambda\alpha = K\alpha \tag{8}$$

for nonzero eigenvalues. Sort the eigenvalues in descending order and use the first $M \leq m$ principal components $\mathbf{V}_k$ ($1 \leq k \leq M$) as the basis vector in $\mathbb{F}$ (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 $\mathbb{F}$. The projection of a point $\mathbf{x}$ onto the k-th kernel principal component $\mathbf{V}_k$ is calculated as

$$\beta_k = (\mathbf{V}_k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{m} \alpha_{k,i} K(\mathbf{x}_i, \mathbf{x}) \tag{9}$$

and $\mathbf{x}$ is represented in the KPCA space as $(\beta_1, \ldots, \beta_M)$.

3 Multi-View Face Detection and Pose Estimation

Assume that a training set of multi-view face examples is provided for the learning. Each example is a windowed grey-level image, possibly preprocessed, denoted by $\mathbf{x} \in \mathbb{R}^N$. See Fig. 1 for some examples. 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. Assume that all left rotated faces (those with view angles between 91° and 180°) are mirrored to right rotated so that the view of every appearance is between 0° and 90°; this does not cause any loss of generality. The view is quantized into a set of L discrete values. Each training face example is assigned into one of L groups according to the nearest view value. This results in L face view training sets, each including various illumination conditions. Another training set of nonface examples is also needed for training face detection.

Now, there are $L + 1$ classes, indexed in the following by $\ell \in \{0, 1, \ldots, L-1\}$ 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).

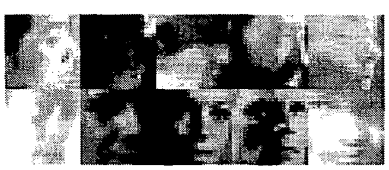

Figure 1. Multi-view face examples.

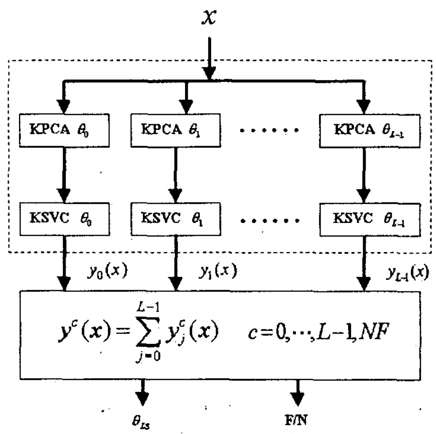

Figure 2. The structure of the multiple KPCA and SVCs and the composite face detector and pose estimator.

3.1 Learning KPCA Subspaces and Classifiers

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

Stage 1 learns a KPCA view-subspace from each of the L view training sets, as defined by the first M most significant components. This effectively performs a nonlinear mapping from the input to the M dimensional KPCA feature space. This stage learning yields L view-subspaces, each determined by a set of support vectors and the corresponding coefficients.

Stage 2 trains a KSVC to classify between L different facial views and nonface pattern in each KPCA view-subspace, using nonlinear features in that subspace computed using a nonface training set as well as the L face view training sets. This $L+1$-class classification effectively performs simultaneous face detection and view (range) estimation. The one-against-the-rest method [31, 8, 39] is used for solving the multi-class problem in a KSVC.

3.2 Face Detection and Pose Estimation

In the testing stage, a windowed pattern is presented to the L KPCA feature extractors, and L feature vectors calculated. The KSVC of each channel then gives an output vector $\mathbf{y}_\ell = (y_\ell^c \mid c = 0, \ldots, L)$, indicating how likely the input belongs to each of the $L+1$ classes. So, L such output vectors $\{\mathbf{y}_\ell \mid \ell = 0, \ldots, L-1\}$ are produced.

The value y_ℓ^c is the evidence of that the input belongs to class c as calculated using the feature in the ℓ-th KPCA view subspace. The evidences from all the L channels are fused by using the sum $y^c(\mathbf{x}) = \sum_{\ell=0}^{L-1} y_\ell^c$. The final decision for the classification is made by maximizing summed 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 nonface subset is given. We choose $L = 10$ for 10 equally spaced angles $\theta_0 = 0°$, $\theta_1 = 10°$, $\cdots$, $\theta_9 = 90°$. The face data is randomly partitioned into three data sets and the nonface data into two data sets for the use in different stages, as shown in Table 4.1. 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.

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

Table 1. Composition of three data sets

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. This is because KPCA learning and projection are where most computational expenses occur in our system. Table 4.2 shows the error rates for various sample size. 500 examples per view are finally adopted to compose Set 1 to balance the tradeoff.

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

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

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 examples 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) shown in Hinton diagrams in Fig. 3. The size of the c-matrices is $(L+1) \times (L+1)$ for the $L+1$ class problem, with the last row and column denoting the nonface class. The entry (i, j) gives the number of samples whose ground truth label (manually labeled, actually, subject to human errors) is class i (in row) and which are classified into class j (in column). The entries are divided by the maximum value of the corresponding row so that the maximum value becomes 1. The sizes of the blocks in the Hinton diagrams are proportional to the normalized values.

Table 4.3 shows the missing and false alarm rates for face detection and Table 4.3 the pose estimation accuracies, all calculated from the c-matrices. There, the $\pm A°$ accuracy is defined as the percentage of samples whose estimated poses are within $\pm A°$ of the ground truth. From these we can see the the kernel method produces much better results than the linear method.

Finally, Fig. 4 shows some examples of face detection and pose estimation from real images. Testing images are collected from VCD movies. Sub-windows of each image are taken from the image at different scales and locations. The pattern in each sub-window is classified into face/nonface, and the pose is estimated if it is a face.

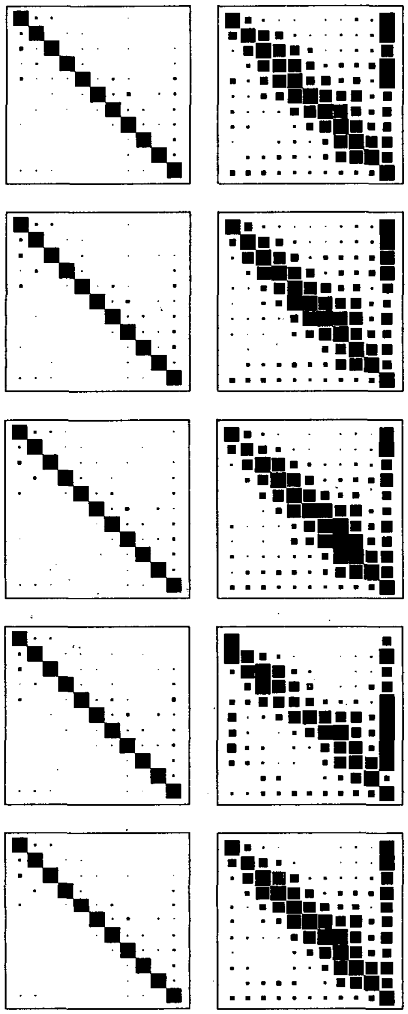

Figure 3. C-matrices in Hinton diagrams for the kernel method (left column) and the linear method (right), calculated based on (top-downwards) the output from individual classifier of 90, 70, 40 and 0 degrees, and the summed output from all the L classifiers.

5 Conclusion

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

Method	Miss (%)	False (%)
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 3. Face Detection Error Rates

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

Table 4. Face Pose Estimation Accuracy

References

[1] Y. Adini, Y. Moses, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):721–732, July 1997.

[2] S. Baker, S. Nayar, and H. Murase. "Parametric feature detection". *International Journal of Computer Vision*, 27(1):27–50, March 1998.

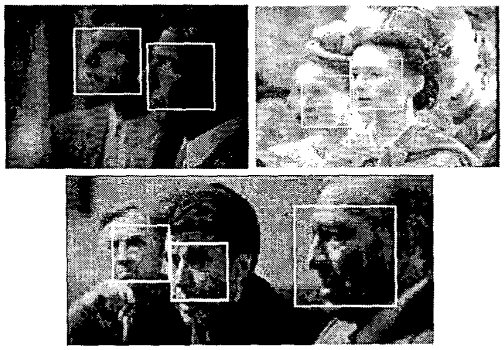

Figure 4. Multi-view face detection results. The estimated views are as follows: From left to right, in the two images on the top, the estimated angles are 10, 0, 60, 50 degrees, respectively; in the bottom image, they are 80, 60, 0 degrees.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[4] P. N. Belhumeur and D. J. Kriegman. "What is the set of images of an object under all possible illumination conditions". *IJCV*, 28(3):245–260, July 1998.

[5] D. Beymer, A. Shashua, and T. Poggio. "Example based image analysis and synthesis". A. I. Memo 1431, MIT, 1993.

[6] M. Bichsel and A. P. Pentland. "Human face recognition and the face image set's topology". *CVGIP: Image Understanding*, 59:254–261, 1994.

[7] C. Bishop and J. M. Winn. "Nonlinear Bayesian image modeling". In *Proceedings of the European Conference on Computer Vision*, pages 1–15, 2000.

[8] 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.

[9] R. Brunelli. Estimation of pose and illuminant direction for face processing. A. I. Memo 1499, MIT, 1994.

[10] H. F. Chen, P. N. Belhumeur, and D. W. Jacobs. "In search of illumination invariants". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I:254–261, 2000.

[11] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

[12] R. Epstein, P. Hallinan, and A. Yuille. "5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models". In *IEEE Workshop on Physics-Based Vision*, pages 108–116, 1995.

[13] K. Etemad and R. Chellapa. "Face recognition using discriminant eigenvectors". 1996.

[14] J. Feraud, O. Bernier, and M. Collobert. "A fast and accurate face detector for indexation of face images". In *Proc. Fourth IEEE Int. Conf on Automatic Face and Gesture Recognition*, Grenoble, 2000.

[15] A. S. Georghiades, D. J. Kriegman, and P. N. Belhumeur. Illumination cones for recognition under variable lighting: Faces. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 52–59, 1998.

[16] 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.

[17] P. W. Hallinan. "A low-dimensional representation of human faces for arbitrary lighting conditions". In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 995–999, 1994.

[18] J. Hornegger, H. Niemann, and R. Risack. "Appearance-based object recognition using optimal feature transforms". *Pattern Recognition*, 33(2):209–224, February 2000.

[19] J. Huang, X. Shao, and H. Wechsler. "Face pose discrimination using support vector machines (SVM)". In *Proceedings of International Conference Pattern Recognition*, Brisbane, Queensland, Australia, 1998.

[20] M. Kirby and L. Sirovich. "Application of the Karhunen-Loeve procedure for the characterization of human faces". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, January 1990.

[21] 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.

[22] 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. Oo Face & Gesture Recognition*, pages 300–305, France, 2000.

[23] B. Moghaddam and A. Pentland. "Probabilistic visual learning for object representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:696–710, July 1997.

[24] 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.

[25] S. Nayar, S. Nene, and H. Murase. Subspace methods for robot vision. *RA*, 12(5):750–758, October 1996.

[26] 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.

[27] E. Osuna, R. Freund, and F. Girosi. "Training support vector machines: An application to face detection". In *CVPR*, pages 130–136, 1997.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] A. Shashua. "On photometric issues in 3d visual recognition from a single 2d image". *International Journal of Computer Vision*, 21:99–122, 1997.

[34] A. Shashua and T. R. Raviv. "The quotient image: Class based re-rendering and recognition with varying illuminations". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):129–139, 2001.

[35] 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.

[36] M. E. Tipping and C. M. Bishop. "Mixtures of probabilistic principal component analyzers". *Neural Computation*, 11(2):443–482, 1999.

[37] 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.

[38] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, New York, 1998.

[39] 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.

[40] 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.

[41] A. Yilmaz and M. Gokmen. "Eigenhill vs. eigenface and eigenedge". In *Proceedings of International Conference Pattern Recognition*, pages 827–830, Barcelona, Spain, 2000.