Support Vector Regression and Classification Based Multi-view Face Detection and Recognition

Yongmin Li, Shaogang Gong and Heather Liddell
Department of Computer Science, Queen Mary and Westfield College
University of London, London, E1 1NS, UK {yongmin, sgg, heather}@dcs.qmw.ac.uk

Abstract

A Support Vector Machine based multi-view face detection and recognition framework is described in this paper. Face detection is carried out by constructing several detectors, each of them in charge of one specific view. The symmetrical property of face images is employed to simplify the complexity of the modelling. The estimation of head pose, which is achieved by using the Support Vector Regression *technique, provides crucial information for choosing the appropriate face detector. This helps to improve the accuracy and reduce the computation in multi-view face detection compared to other methods. For video sequences, further computational reduction can be achieved by using* Pose Change Smoothing *strategy. When face detectors find a face in frontal view, a* Support Vector Machine *based multi-class classifier is activated for face recognition. All the above issues are integrated under a* Support Vector Machine *framework. Test results on four video sequences are presented, among them, detection rate is above 95%, recognition accuracy is above 90%, average pose estimation error is around 10° , and the full detection and recognition speed is up to 4 frames/second on a PentiumII300 PC.*

1. Introduction

In recent years, significant progress has been made in the area of face detection and recognition. However, most of the previous work in face detection is limited to the frontal view. Sung and Poggio proposed a *Neural Network* (NN) based approach which uses 6 face and 6 nonface prototypes to build the hidden layers. The distances between a detected pattern and each of the 12 prototypes are measured to synthesize the final output [12]. Another NN based approach proposed by Rowley *et al.* can cope with the rotation in the image plane by designing an extra NN to estimate the orientation of face [9]. Osuna *et al.* presented a *Support Vector Machine* (SVM) based approach for frontal view face detection. This is one of the first applications of SVM on real-world problems [7]. How to deal with the rotation in depth, i.e. detect faces across views, however, remains a challenging problem.

The issue of face recognition has also been extensively addressed during the past decade [5, 13, 3]. Among them, the eigenface approach proposed in [5, 13] uses Principal Component Analysis (PCA) to code face images and capture face features. This approach has then been extended to view-based and modular eigenspaces intended for recognising faces under varying views. [3] uses similarity vectors to estimate head pose and recognise faces across views. However, the results on multi-view face recognition are usually inferior to frontal view face recognition.

We propose an approach which uses SVM for multi-view face detection and recognition. On the detection aspect, this approach can cope with face rotation in depth by using multiple face detectors specific to different views. On the recognition aspect, pose estimation results are employed to activate frontal-view face recognisers. An important characteristic of our approach is that it can obtain a robust performance in a poorly constrained environment, especially for low resolution, large scale changes, and rotation in depth.

The paper is organised as follows: The basic notion and properties of SVM are introduced in Section 2. Our approach to multi-view face detection and face recognition using SVM is described in Section 3. Section 4 provides experiment results. The conclusions are drawn in Section 5.

2. Support Vector Machine

The SVM is based on *Structural Risk Minimization* theory [1, 15, 4]. For given observations $\mathbf{x}$ and interpretations y, one finds the optimal approximation

$$f(\mathbf{x}, \alpha) = \mathbf{w} \cdot \Phi(\mathbf{x}) + b \tag{1}$$

where α represents the parameters of a learning machine, Φ is a map from the original data space of $\mathbf{x}$ to a high-dimensional feature space and b is the threshold [15]. If the interpretation y only takes values -1 and $+1$, the learning problem is referred to as *Support Vector Classification* (SVC). Otherwise, if the domain of y contains continuous real values, it is *Support Vector Regression* (SVR).

By introducing a kernel function

$$K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}), \tag{2}$$

the SVC problem can be transformed to maximize

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{3}$$

$$\text{subject to} \quad \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{4}$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, l \tag{5}$$

which gives a separating function

$$f(\mathbf{x}) = \sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}, \mathbf{x_i}) + b \quad (6)$$

On the other hand, the SVR problem can be solved by maximizing

$$W(\alpha^*, \alpha) = -\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(\mathbf{x_i}, \mathbf{x_j}) - \varepsilon \sum_{i=1}^{l} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i (\alpha_i^* - \alpha_i) \quad (7)$$

$$\text{subject to} \quad \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) = 0 \quad (8)$$

$$0 \leq \alpha_i^*, \alpha_i \leq C \quad (9)$$

which provides the solution

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x_i}) + b \quad (10)$$

It is interesting to notice that only a few parameters α take non-zero values, i.e. only those "important" examples, known as *Support Vectors* (SVs), are selected to construct the optimal approximation functions (6) and (10). Functions (6) and (10) are linear combinations of the SVs in high-dimensional feature space. However, instead of computing the map Φ explicitly, one only needs to compute the kernel function (2) with great ease.

There have been many applications in the area of SVC, such as face detection [7], text information categorization [4] and Optical Character Recognition [10]. However, most of the published results on SVR are still on toy problem [2, 11]. In the following sections, our approach of combining SVC, SVR and multi-class SVC for multi-view face detection and recognition is described in detail.

3. Multi-view Face Detection and Recognition under a SVM Framework

Our approach to multi-view face detection and recognition using SVM can be described as follows:

1. using motion and skin colour context to bootstrap sub-images containing faces;
2. exhaustively scanning the sub-images with different scales;
3. for each image patch from the scanning, estimating the "pose" using SVR pose estimators;
4. choosing a proper face detector from a set of SVC based multi-view face detectors according to the estimated "pose" to determine whether or not the pattern is a face. If the output of the face detector is above a preset threshold, then a face is detected, and the position, scale and pose of the detected face are fed to the next step of recognition. Otherwise, the pattern is regarded as a meaningless patch of image;
5. synthesizing all detections to a single detection;
6. if the detected face is in frontal view, a SVC based face recogniser is activated to perform recognition on the face.

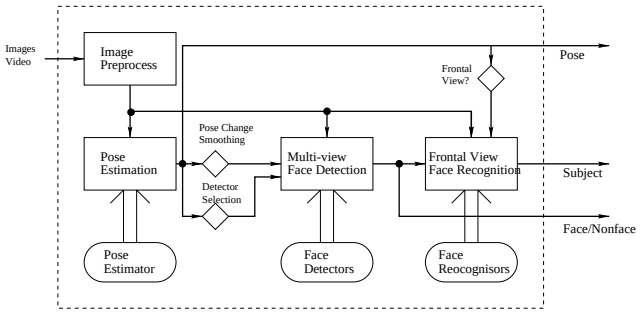

Figure 1. *Multi-view face detection and recognition.*

The process is illustrated in Figure 1. It is worth noticing that head pose estimation plays an important role in the process for reducing computational cost, improving accuracy, and providing useful information for visual interaction. These topics will be presented in detail in the following sections.

3.1. Estimating Head Pose Using SVR

Face images of one person in different poses can appear much more dissimilar than faces of different people at the same pose. However, if the pose of a face image is known, then the recognition problem can be simplified to a great extend. Also, pose information provides a useful cue for motion prediction, selected object tracking, and intention understanding. which are crucial in visual interaction.

In our approach, a SVR based pose estimator is trained using 1596 face images which can give robust estimation with only a small number of SVs.

Two Sobel operators (horizontal and vertical) are adopted as a filter to preprocess the training face images. The two filtered images are combined together as the compositive patterns (see Figure 2). As the filter captures the changes both in horizontal and vertical directions which correspond to yaw and tilt changes respectively, the filtered patterns are more representative than the original images.

PCA [8, 6] is performed on the filtered patterns in order to reduce the dimensionality of the training examples. Figure 2 illustrates the first 10 Principal Components (PCs) and the reconstructed patterns from the first 20 PCs compared to the original images and the filtered patterns.

Two SVR based pose estimators, one for yaw and the other for tilt, are constructed to estimate the head poses. For each of the pose estimators, the preprocessed images are taken as patterns x, and yaw/tilt angles as y are fed into a decomposition algorithm based on the LOQO [14], an algorithm for the quadratic optimization problem. In most cases, the proportion of SVs is only 10-15% of the training examples.

When running the pose estimators on an image (or a sequence of images) containing faces, exhaustive scanning on a segmented image region (normally obtained by using motion and skin colour) is performed. The estimated result of each cropped sub-image is fed to the multi-view face detectors to determine whether it is a face.

Figure 2. *Representation for pose estimation. From top row to bottom row are the original face images, the corresponding filtered patterns, reconstructed patterns from the first 20 PCs, and the first 10 significant PCs.*

3.2. Multi-view Face Detection

Face detection can be treated as a classification problem, i.e. separating face patterns from nonface patterns. There are basically three methods to perform multi-view face detection. A straightforward way is to build a single detector dealing with all views of a face. The second approach is to build several detectors, each of them corresponding to a specific view. When detecting, all the detectors are employed, and if one or more of them give positive output, then a face is considered to be detected. Our preliminary experiments showed that the first method led to poor performance due to serious nonlinear variations of faces between different views. The second approach performs better as expected but the computation is expensive, as each of the multi-view face detectors is calculated on a given pattern. We adopted the third approach to the problem by estimating the "pose" of a given pattern before choosing only one of the multi-view face detectors to determine whether the targeted image pattern is a face.

As shown in Figure 3, faces in different views are divided into 8 segments: left profile, left frontal, right frontal, right profile in the horizontal direction (yaw), and up, down in the vertical direction (tilt). We build the multi-view face models according to three considerations:

1. Faces are symmetrical along the vertical line across the nose-bone, so the right view faces can be converted to left view without losing the general face characteristics. Based on this, one only needs to model the multi-view faces either in the left or the right view. As illustrated in Figure 3, only four detectors are constructed.

2. Soft boundaries between segments, i.e. overlap by 10° between neighbouring segments, are introduced to provide seamless detection.

3. The vertical separating angle is 90° in tilt, and horizontal 45° in yaw with expectation to separate one-eye faces (profile) from two-eye faces (frontal) effectively.

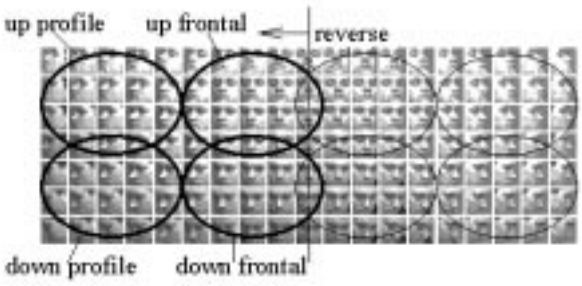

Figure 3. *Modelling multi-view faces. Four detectors are modelled based on the symmetry property of human face: up profile, up frontal, down profile, down frontal. The neighbour segments are overlapped, aiming to provide seamless detection across views.*

When training each of the multi-view face detectors, the face images corresponding to the specific view are selected from the same database for pose estimation as positive examples (faces). Negative examples (non-faces) are collected by a boot-strapping technique [12] from a set of scenic pictures.

The original size of the example images is 20x20 in pixels. The method described in Section 3.1 is used for image preprocessing. The PCA technique is employed to reduce the dimensionality of training examples and to extract features (general face related). Because we are only interested in detecting faces, the same PCA result as Figure 2 was employed which was trained using only positive images (faces).

A decomposition algorithm (SVC, different from the decomposition algorithm in Section 3.1) is developed to train the SVM face detector, where the LOQO algorithm is employed to solve each decomposed subproblem.

Provided the pose of an image patch is known, the computation of face detection can be greatly reduced by only choosing the appropriate face detector for the given pose. This strategy can be efficiently employed in both static images and video sequences.

For video sequences, computational cost can be further reduced by applying a *Pose Change Smoothing* strategy. Because pose change between two continuous frames is usually small, one can make an assumption that it should be below a threshold θ, for example 20° , i.e.

$$\|tilt(t) - tilt(t-1)\| < \theta \quad (11)$$
$$\|yaw(t) - yaw(t-1)\| < \theta \quad (12)$$

where $tilt(t), yaw(t)$ are the current pose estimates, $tilt(t-1)$ and $yaw(t-1)$ are the previous pose estimates. If the given pose does not satisfy the above conditions, then the image patch can be regarded as not containing a face and face detectors are not activated. In the case where no previous detection is available, one can skip *Pose Change Smoothing* to increase the possibility of detection.

3.3. Face Recognition in Frontal Views

Face recognition can be regarded as a multi-class classification problem, i.e. defining several classes, each corresponding to a subject involved in the recognition task, and determining into which class a given face image should fall.

If faces can be tracked continuously in a dynamic visual interactive environment, then it is not necessary to continue face recognition on each frame. Furthermore, because the change of face appearance in frontal view is smoother than in profile views, i.e. less change in frontal view than in profile views with the same rotation in depth, we only construct a face recogniser in frontal view.

In our approach, m SVM classifiers are trained, based on an one-against-all strategy, where m is the number of subjects needed for the recognition task. When recognising a new face image, all the m classifiers give their own outputs, and the final output is generated by synthesizing the m outputs from different classifiers.

Suppose there are m subjects needed to be recognised, where $l = \sum_{i=1}^{m} l_i$ is the number of training face examples, and l_i is the number of training face examples of subject i. One can build m SVM classifiers c_k. For each $c_k, k = 1, 2, ..., m$, the l_k examples are chosen as positive examples, and all the other $\sum_{i=1, i\neq k}^{m} l_i$ examples as negative examples. After training using the method described in Section 2, one obtains m classifiers:

$$c_k = \sum_{i=1}^{l} \alpha_{ki} y_i K(x_i, x) + b_k \tag{13}$$

When recognising a new face image, each of the m classifiers give their own outputs c_i. There are three possibilities for the output set c_i:

1. Only one has a positive value, and all the others are negative. This is the ideal case. The identification number of the positive classifier is selected as the final output.

2. More than one of the outputs are positive. If we make the assumption that a high output value leads to a high likelihood a face belongs to a subject, then we can adopt the maximal value.

3. None of the outputs is positive. That should be regarded as the new face belongs to a subject who is not included in our subject set. Thus the final output may be taken as "new subject".

In our experiments, we impose another assumption that a test face image must belong to a subject in the recognition set. Under this assumption, the maximal output can be adopted as the final result in the case of (3). To summarise, the recognised result can be achieved by:

$$id = argmax_{i=1}^{m} c_i \tag{14}$$

where id is the identification number of the result subject.

4. Experiments and Discussions

4.1. Pose Estimation

Two pose estimators were trained to estimate the yaw and tilt angles of face images respectively. The training set consisted of 1596 20x20 images taken from 12 subjects, one image per pose for each subject containing poses from 0° - 180° in yaw and 60° -120° in tilt with intervals of 10° . All images were filtered by horizontal and vertical Sobel operators before PCA was employed on the compositive patterns to reduce the dimension. Before the patterns were fed to the training algorithm, a simple normalization was carried out to make the deviation of each pattern to 1. Another set of 1283 images were used to test the generalization performance of the trained pose estimators.

The effect of using different numbers of significant PCs, i.e. the dimension of the preprocessed patterns, is shown in Table 1. The results depict that low dimensional PCA representation (20-30 in dimension for example) can provide satisfactory performance in pose estimation. In most cases, the proportion of SVs is only 10-15%.

dim	yawSV	tiltSV	trainT	testT	errYaw	errTilt
5	535	164	935+76	17	15.48	9.77
10	284	99	224+50	16	12.12	7.67
15	257	97	168+43	18	12.36	8.64
20	241	108	116+56	18	11.06	8.69
25	211	129	78+55	19	11.11	8.67
30	225	130	115+55	21	10.23	8.78
35	216	140	103+46	22	10.55	8.84
40	220	149	93+58	23	10.73	8.85
45	234	163	107+59	25	10.63	8.93
50	247	163	120+70	26	9.02	10.51
55	249	168	124+62	27	10.54	8.98
60	246	173	125+75	28	10.43	9.01
65	254	176	101+60	29	10.37	9.03
70	253	174	149+74	30	10.23	9.05
75	256	181	127+75	32	10.22	9.01
80	262	180	117+61	32	10.22	9.03
85	267	182	117+76	33	10.21	9.04
90	269	186	156+63	35	10.20	9.05
95	266	186	155+63	36	10.17	9.05
100	274	186	136+65	37	10.20	9.06

Table 1. *Performance of pose estimators on different dimension of preprocessed data. From left to right, numPC: dimension of PCA space, yawSV: SV number of yaw estimator, tiltSV: SV number of tilt estimator, trainT: training time in second (yaw+tilt), testT: time in second used on testing 1283 new images, errYaw: average error in yaw, errTile: average error in tilt. Parameters of SVR are kernel: Gaussian ($2\sigma^2 = 1$), C: 1000, ε: 20, training data; 1596, test data: 1283.*

4.2. Face Detection

We built four SVC based face detectors for up profile view, down profile view, up frontal view and down frontal view as shown in Figure 3. The training set included

1596 face images and nonface images obtained by a bootstrapping process. Before training, the right view face images (with 90-180° in yaw) were converted along the middle vertical line to left view images. Then the images were projected to the first 20 significant PCs (PCA is trained on all 1596 images) and normalized with deviation equal to 1. Table 2 lists the results of training.

detector	yaw	tilt	numTrain	numSV
up profile	0-40,130-180	60-90	480+500	169
down profile	0-40,130-180	90-120	480+500	159
up frontal	50-90,90-130	60-90	480+500	188
down frontal	50-90,90-130	90-120	480+500	152

Table 2. *Training results of face detection. After making use of the symmetrical property of the face images, only four detectors are trained: up profile, down profile, up frontal, down frontal. From left to right, detector: name of face detector, yaw/tilt: view in charge, numTrain: number of training patterns, numSV: number of SVs obtained.*

4.3. Face Recognition

Face recognition was carried out on a small set of subjects including 10 people. The training set included 90 face images with 9 for each subject. All the faces were collected in frontal view or near frontal view (only 10° rotation off the image plane). To make the face images consistent to the face detector, we ran the detector on those images, then cropped the detected patches as the real training examples for face recognition. Part of the original face images and the corresponding cropped parts by the detector are shown in Figure 4. After training on the 90 cropped images, a face recogniser consisting of 10 one-against-all classifiers was built. Table 3 lists the training results.

Figure 4. *Sample training faces for face recognition. All faces are in frontal or near frontal view (10° rotation off the image plane both for horizontal and vertical direction).*

4.4. Integrated Multi-view Face Detection and Recognition

All the modules for head pose estimation, face detection, and face recognition were integrated into a SVM based system. First, the regions containing faces were segmented from the whole image. The second step was to scan the segmented region with different scales. Pose estimators were employed on each of the image patches. The *Pose Changing Smoothing* strategy was used to activate the multi-view face detectors for better real-time performance. After scanning, the maximum detection, as well as its pose, position, scale were saved as the final detection. If the detected face was frontal view, the recogniser is turned on to perform recognition.

Table 4 lists some results from the experiment of face detection and recognition on four test sequences. The full detection and recognition speed is up to 4 frames/second on a PentiumII300 PC. A sample frame from one sequence with the pose estimation, face detection and recognition results is shown in Figure 5.

It is important to point out that the face detectors also implicitly performed the task of alignment for recognition. On a small group of subjects such as 10 in our experiments, recognition accuracy based on such alignment is rather good. However, if the number of subjects to be recognised increases, recognition accuracy based on such alignment would decrease. More complex methods are needed for the scalability of recognition.

seq	frame	errT	errY	detected	frontal	recognised	time
1	100	10.8	5.3	100(100%)	36	32(89%)	26
2	200	11.8	10.2	198(99%)	64	61(95%)	56
3	200	8.8	16.2	200(100%)	56	53(95%)	51
4	200	8.0	7.0	193(97%)	101	93(92%)	57
total	700	9.7	10.3	691(99%)	257	239(93%)	190

Table 4. *Test results on four sequences. From left to right, seq: sequence number, frame: number of frames in sequence, errT: average absolute error of tilt, errY: average absolute error of yaw, detected: number of frames where faces were detected, frontal: number of frames where faces are in frontal view (recognition is performed), recognised: number of frames where subjects were correctly recognised, time: total detection and recognition time in second.*

5. Conclusions

An integrated SVM approach to multi-view face detection and recognition is presented in this paper. The contributions of our work include:

1. Using SVR to construct pose estimators which provides robust and fast estimation of head pose.

2. A set of SVC classifiers were trained for multi-view face detection. By using the symmetrical property of the face images, those classifiers can be built with only half of the possible views.

3. Pose information was employed to guide the selection of face detectors so as to improve the detection accuracy, and at the same time, to ease computation in detection. Further computational reduction was achieved by using the *Pose Change Smoothing* technique.

4. When faces were in frontal view which can be determined by estimated pose information, a SVC multi-class classifier was activated to perform face recognition. Because the appearance change of faces in frontal

subID	1	2	3	4	5	6	7	8	9	10	total
SVs	27	16	16	25	16	13	19	23	14	17	83
%	30.00	17.78	17.78	27.78	17.78	14.44	21.11	25.56	15.56	18.89	92.22

Table 3. *Face recognition training results. The preprocessed data dimension is 30 in this experiment. Ten recognisers were trained based on an one-against-all principle. For each recogniser, nine positive and 81 negative training examples were used. subID: subject identification number, SVs: number of SVs. The last row shows the percentage of SVs from the training set. The last column shows the summarized results: the number of all unique SVs and their proportion to the total training examples.*

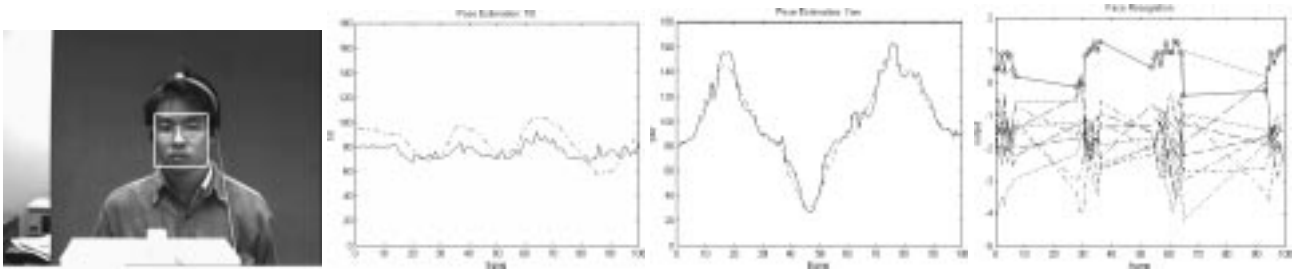

Figure 5. *A sample frame from a test sequence together with pose estimation (tilt/yaw) and recognition on the sequence. The detection found by the SVM face detector are marked with white boxes. The output values of each classifier of the face recogniser are shown on the right hand, where the outputs of the classifier corresponding to the test subject are marked with crosses. A simple maximum method is used to determine the identification of subject.*

view is more smooth than in other views, this method provides a robust recognition performance.

5. All the above issues were integrated in a SVM based framework.

Test results on four live video sequences containing faces across views were also presented, among which, the detection rate is above 95%, the recognition accuracy is above 90%, and the pose estimation in both yaw and tilt is around 10° . The full detection and recognition speed was up to 4 frames/second. However, the current work can be further improved by using temporal context to reduce the searching space for face detection. Run-time performance can also be improved by refining the SV set.

Acknowledgments

The authors wish to thank A. Smola who provided the LOQO program.

References

[1] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, pages 1–47, 1998.

[2] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA, 1997.

[3] S. Gong, E.-J. Ong, and S. McKenna. Learning to associate faces across views in vector space of similarities to prototypes. In *Proc. British Machine Vision Conference*, Southampton, England, 1998.

[4] M. Hearst, B. Scholkopf, S. Dumais, E. Osuna, and J. Platt. Trends and controversies – support vector machines. *IEEE Intelligent Systems*, 1998.

[5] B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. *Automatic Systems for the Identification and Inspection of Humans*, 1994.

[6] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. In *IJCV*, 1995.

[7] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. Computer Vision and Pattern Recognition'97*, pages 130–136, 1997.

[8] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE CVPR*, Seatle, 1994.

[9] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 1998.

[10] B. Scholkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.

[11] A. Smola, B. Scholkopf, and K.-R. Muller. General cost functions for support vector regression. In T. Downs, M. Frean, and M. Gallagher, editors, *Proc. of the Ninth Australian Conf. on Neural Networks*, pages 79–83, Brisbane, Australia, 1998.

[12] K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical report, Massachusetts Institute of Technology, 1994. A.I. MEMO 1521.

[13] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991.

[14] R. Vanderbei. Loqo: An interior point code for quadratic programming. Technical report, Princeton University, 1994. Technical Report SOR 94-15.

[15] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, New York, 1995.