Available online at www.sciencedirect.com

SCIENCE DIRECT®

Image and Vision Computing 24 (2006) 615–625

www.elsevier.com/locate/imavis

Dynamics of facial expression extracted automatically from video

Gwen Littlewort *, Marian Stewart Bartlett, Ian Fasel, Joshua Susskind, Javier Movellan

Institute for Neural Computation, University of California, Diego San Diego, CA 92093-0523, USA

Received 1 December 2004; received in revised form 15 August 2005; accepted 15 September 2005

Abstract

We present a systematic comparison of machine learning methods applied to the problem of fully automatic recognition of facial expressions, including AdaBoost, support vector machines, and linear discriminant analysis. Each video-frame is first scanned in real-time to detect approximately upright-frontal faces. The faces found are scaled into image patches of equal size, convolved with a bank of Gabor energy filters, and then passed to a recognition engine that codes facial expressions into 7 dimensions in real time: neutral, anger, disgust, fear, joy, sadness, surprise. We report results on a series of experiments comparing spatial frequency ranges, feature selection techniques, and recognition engines. Best results were obtained by selecting a subset of Gabor filters using AdaBoost and then training Support Vector Machines on the outputs of the filters selected by AdaBoost. The generalization performance to new subjects for a 7-way forced choice was 93% or more correct on two publicly available datasets, the best performance reported so far on these datasets. The outputs of the classifier change smoothly as a function of time and thus can be used for unobtrusive expression dynamics capture. We developed an end-to-end system that provides facial expression codes at 24 frames per second and animates a computer-generated character. In real-time this expression mirror operates down to resolutions of 16 pixels from eye to eye. We also applied the system to fully automated facial action coding.

© 2005 Elsevier B.V. All rights reserved.

Keywords: Facial action coding; Support vector machines; Facial expression; Adaboost

1. Introduction

We present results on a user independent fully automatic system for real time recognition of basic emotional expressions from video. The system automatically detects frontal faces in the video stream and codes each frame with respect to seven dimensions: Neutral, anger, disgust, fear, joy, sadness, surprise. We conducted empirical investigations of machine learning methods applied to this problem, including comparison of recognition engines, feature selection techniques, spatial frequency ranges, and methods for multiclass decisions with binary classifiers. Best results were obtained by selecting a subset of Gabor filters using AdaBoost and then training Support Vector Machines on the outputs of the filters selected by AdaBoost. The combination of AdaBoost and SVM's enhanced both speed and accuracy of the system. The system presented here is fully automatic and operates in real-time at a high level of accuracy (93% generalization to new subjects on a 7-alternative forced choice).

2. Facial expression data

The facial expression system was trained and tested on Cohn and Kanade's DFAT-504 dataset [15]. This dataset consists of 100 university students ranging in age from 18 to 30 years. 65% were female, 15% were African-American, and 3% were Asian or Latino. Videos were recoded in analog S-video using a camera located directly in front of the subject. Subjects were instructed by an experimenter to perform a series of 23 facial expressions. Subjects began and ended each display with a neutral face. Before performing each display, an experimenter described and modeled the desired display. Image sequences from neutral to target display were digitized into 640 by 480 pixel arrays with 8-bit precision for grayscale values.

For our study, we selected the 313 sequences from the dataset that were labeled as one of the 6 basic emotions. The sequences came from 90 subjects, with 1–6 emotions per subject. The first and last frames (neutral and peak) were used as training images and for testing generalization to new subjects, for a total of 625 examples. The trained classifiers were later applied to the entire sequence.

* Corresponding author. Tel.: +858 822 5241.
E-mail address: gwen@mplab.ucsd.edu (G. Littlewort).

0262-8856/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2005.09.011

2.1. Real-time face detection

We developed a real-time face detection system that employs boosting techniques in a generative framework [11] and extends work by [28]. Enhancements to [28] include the use of continuous non-linear transfer functions rather than binary threshold functions, smart feature search, and a novel cascade training procedure, combined in a generative framework. Source code for the face detector is freely available at http://mplab.ucsd.edu. The face detector was trained on 5000 faces and millions of non-face patches from about 8000 images collected from the web by Compaq Research Laboratories. Accuracy on the CMU-MIT dataset, a standard public data set for benchmarking frontal face detection systems, is 90% detections and 1/million false alarms, which is state-of-the-art accuracy. The CMU test set has unconstrained lighting and background. With controlled lighting and background, such as the facial expression data employed here, detection accuracy is much higher. The system presently operates at 24 frames/s on a 3 ghz Pentium IV for 320×240 images.

All faces in the DFAT-504 dataset were successfully detected. The automatically located faces were rescaled to 48×48 pixels. The typical distance between the centers of the eyes was roughly 24 pixels. Many other approaches to automatic facial expression recognition include explicit detection and alignment of internal facial features. We found that further registration based facial features was not necessary for this task, thus providing considerable savings in processing time, without sacrificing performance. The images were converted into a Gabor magnitude representation, using a bank of Gabor filters at 8 orientations and 5 spatial frequencies (4:16 pixels per cycle at 1/2 octave steps) [18].

3. Facial expression classification

We first examined facial expression classification based on support vector machines (SVM's). SVM's are well suited to this task because the high dimensionality of the Gabor representation O(10^5) does not affect training time, which depends only on the number of training examples O(10^2). The system performed a 7-way forced choice between the following emotion categories: Happiness, sadness, surprise, disgust, fear, anger, neutral.

3.1. Strategies for multiclass decisions with SVM's

Support vector machines make binary decisions. There are a number of methods for making multiclass decisions with a set of binary classifiers. (See [14] for a review). Here, the seven-way forced choice for six emotions plus neutral was trained in two stages. In stage I, support vector machines performed binary decision tasks. We explored three approaches to training binary decisions: one-versus-one, one-versus-all, and all possible partitions. Stage II converts the representation produced by the first stage into a probability distribution over the seven expression categories. To this effect, we have implemented and evaluated several approaches: *K*-nearest neighbor, a simple voting scheme, and multinomial logistic ridge regression.

3.1.1. Partitioning into binary decisions

There are a number of strategies for partitioning the classification task into binary decisions. The simplest strategy is to train 1 versus all. Pairwise partitioning strategies have been advocated by [17] and [24], whereas others (e.g. [7]) advocate exploring the space of all possible partitions.

For pairwise partitioning (1:1), SVM's were trained to discriminate all pairs of emotions. For seven categories that makes 21 SVM's. In 1:1 partitioning, the number of training samples for each SVM may be relatively small. If some subjects performed some expressions and not others, as in this dataset, identity signals can interfere with the learning of expression. To avoid this, we trained on identity-matched pairs, where for example, the happy vs. surprise SVM is trained on only those subjects who gave samples of both happiness and surprise. An alternative to training SVM's to discriminate each pair of emotions was to train SVM's to discriminate one emotion from everything else (1:all). This strategy employed a larger number of training examples, 626, which diluted identity effects. An extension of the 1:all approach was to consider all possible non-trivial binary partitions of the classes. With 7 classes, there are seven 1:6 classifiers, twenty one 2:5 classifiers and thirty five 3:4 classifiers.

3.1.2. Combining outputs of multiple binary classifiers

In the system presented here, the SVM outputs were combined to make a 7 alternative forced choice. The most common way to combine SVM outputs for multiclass decisions is by voting. This procedure counts the number of stage 1 classifiers aligned with each emotion. For example, if one SVM indicates happiness and not surprise, happiness gets $+1$ and surprise gets -1. These votes are summed over all of the SVM's. Softmax ensures each class is allocated a number between 0 and 1, with unit sum over classes. We also explored a variation on voting which uses the sum of the classifier margins, which are typically clustered around $+1$ or -1, instead of the binary outputs. This variation made little difference, and the voting results presented here use binary outputs.

We compared voting to nearest neighbor, and to a learned mapping based on multinomial logistic ridge regression (MLR) . In nearest neighbor, the continuous SVM output (the margin) for each of the n SVM's gives an n-dimensional pattern vector. The test image is assigned the class of the training image with the shortest Euclidean distance between their pattern vectors. MLR learns the weight matrix that maps the outputs of Stage one classifiers onto the seven emotions. MLR is a maximum likelihood approach, which is equivalent to a single layer perceptron with weight decay and with SoftMax competition between the outputs. SVM's are in some sense the limiting case of MLR as the ridge term goes to zero. The regression was implemented using the Newton–Raphson method and a ridge term coefficient of 0.001. The advantage of this data-dependent

Table 1
Comparison of strategies for multiclass decisions using SVM's

		Nnbr	Voting	MLR
Linear	1:1	82.7	81.6	85.8
SVM's	1:all	81.6	86.2	87.5
	all poss.	83.0	87.2	89.4
Nonlinear	1:1	83.2	82.9	86.1
SVM's	1:all	81.4	88.0	89.8
	all poss.	85.1	89.9	90.4

second stage is that it could learn common confusions and biases, which lead to errors in a direct voting situation.

Generalization results to novel subjects was tested using leave-one-subject-out cross-validation. Results are given in Table 1. Linear, polynomial, and RBF kernels with Laplacian and Gaussian basis functions were explored. Linear and Gaussian RBF kernels performed best and are presented here. The latter showed very low sensitivity to Gaussian width parameter $\sigma \approx$ root mean square pair-wise distances. The soft margin approach, allowing some training examples to lie within the margin, was not used, since it was not found to improve generalization.

For Stage I, partitioning the classification task into binary decisions, 1:all usually outperformed 1:1 partitioning, and all possible partitions gave the best performance. Of the Stage II strategies for combining the outputs of multiple SVM's into a 7-way forced choice, MLR was substantially better than nearest neighbor (5.3 percentage points). Voting was slightly but consistently less effective than MLR, typically 1.3 percent for 1:all and all partitions.

For the comparisons in the subsequent sections, 1:all partitioning followed by voting was employed due to training speed. The optimal strategies determined in this section (all possible partitions and MLR) will be reintroduced in the final system.

3.2. SVM's and Adaboost

SVM's and Adaboost are both well suited to the tasks described in this paper, because they can cope with very large representation space, they generalize well, perform decisions in real-time and are simple to train. Here we review the similarities between SVM's and Adaboost, as well as show where the two algorthms diverge.

SVM's and Adaboost are both large margin classifiers. The two approaches can be thought of as maximizing a margin that depends on the weights α and the hypotheses h [12], although Adaboost does not usually attain the maximum. In both classifiers, the margin is of the following form:

$$\max_{\alpha} \min_{i} \frac{(\alpha.\mathbf{h}(x_i))y_i}{||\alpha||||\mathbf{h}(\mathbf{x_i})||} \tag{1}$$

where x_i and y_i are the input and label for training example i.

One difference between the classifiers is that the norms in the demoninator of Eq (1) are 2-norms for the standard form of the SVM, whereas for Adaboost there is a 1-norm of α and an infinity norm of h. In a high dimensional space, these differences could lead to large differences in performance.

There are theoretical upper bounds on the generalization errors for these classifiers, [4,12]. In practice however, the empirical errors are well below these bounds and the theoretical limit does not predict which classifier will work best in a given applied setting.

The two approaches both concentrate on borderline examples, examples that are more difficult to classify, although they may differ somewhat in which examples are considered borderline. In SVM's the decision boundary is defined by those training examples which are closest to the separating hyperplane, and which can be thought of as the most difficult to classify. In Adaboost, the misclassified examples are boosted relative to the other training examples during learning. However, SVM selects particular examples (support vectors) while adaboost selects features.

We compared SVM's to Adaboost on the task of emotion classification. The features employed for the emotion classifier were the individual Gabor filters. There were 5 spatial scales (4:16 pixels per cycle), 8 orientations, and 48×48 image locations, giving 5×8×48×48=92, 160 possible features. A subset of these features was chosen using Adaboost. On each training round, the Gabor feature with the best classification performance for the current boosting distribution was chosen. The performance measure was a weighted sum of errors on a binary classification task, where the weighting distribution (boosting) was updated at every step to reflect how well each training vector was classified. Adaboost had an external parameter consisting of the number of training rounds for each emotion. We chose this parameter such that there was no training error and the generalization error was flat. (see Fig. 1). The union of all features selected for each of the seven emotion classifiers resulted in a total of 538 features. When we increased the number of the number of frequencies from 5 to 9 in Section 3.3, the total number of selected features was 900.

Classification results are given in Table 2. The generalization performance with Adaboost was comparable to linear SVM performance. Adaboost had a substantial speed advantage, as shown in Table 3. There was a 170-fold reduction in the number of Gabor filters used. The convolutions were calculated in pixel space, rather than Fourier space, which

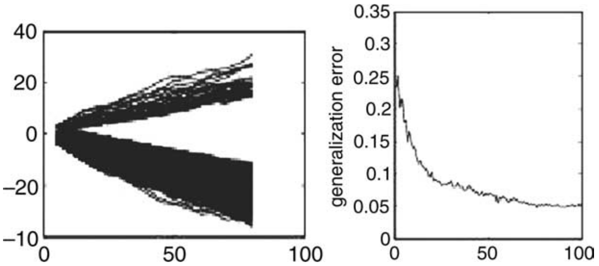

Fig. 1. Stopping criteria for Adaboost training. (a) Output of one expression classifier during Adaboost training. The response for each of the training examples is shown as a function of number features as the classifier grows. (b) Generalization error as a function of the number of features chosen by Adaboost. Generalization error did not increase with 'overtraining'.

Table 2
Leave-one-out generalization performance of Adaboost, SVM's and AdaSVM's (48×48 images)

ω	kernel	Adaboost	SVM	AdaSVM
4:16	Linear	87.2	86.2	88.8
4:16	RBF		88.0	90.7
2:32	Linear	90.1	88.0	93.3
2:32	RBF		89.1	93.3

ω: Gabor wavelength range, sampled at 0.5 octave intervals.

Table 3
Processing time and memory considerations

	SVM		Adaboost	AdaSVM	
	Lin	RBF		Lin	RBF
Time t	T	90 t	0.01 t	0.01 t	0.0125 t
Time t′	T	90 t	0.16 t	0.16 t	0.2 t
Memory	M	90 m	3 m	3 m	3.3 m

Time t' includes the extra time to calculate the outputs of the 538 Gabors in pixel space for Adaboost and AdaSVM, rather than the full FFT employed by the SVM's.

reduced the advantage of feature selection, but it nevertheless resulted in a substantial speed benefit.

3.3. Combining feature selection by Adaboost with classification by SVM's

SVM's have been shown to perform better when the feature space is dense, meaning that each feature is highly relevant to the problem [25]. We explored training SVM classifiers on the features selected by Adaboost. Adaboost is not only a fast classifier; it is also a feature selection technique. An advantage of feature selection by Adaboost is that features are selected contingent on the features that have already been selected. In feature selection by Adaboost, each Gabor filter is a treated as a weak classifier. Adaboost picks the best of those classifiers, and then boosts the weights on the examples to weight the errors more. The next filter is selected as the one that gives the best performance on the errors of the previous filter. At each step, the chosen filter can be shown to be uncorrelated with the output of the previous filters [13,12].

SVM's were trained on the features selected by Adaboost. When we trained SVM's on the thresholded outputs of the selected Gabor features, they performed no better than Adaboost. However, we trained SVM's on the continuous outputs of the selected filters. We informally call these combined classifiers AdaSVM. AdaSVM's outperformed straight Adaboost by 3.8 percent points, a difference that was statistically significant ($z=1.99$, $P=0.02$). AdaSVM's outperformed SVM's by an average of 2.7 percent points, an improvement that was marginally significant ($z=1.55$, $P=0.06$). Adaboost and AdaSVM's are much faster in application (Fig. 2).

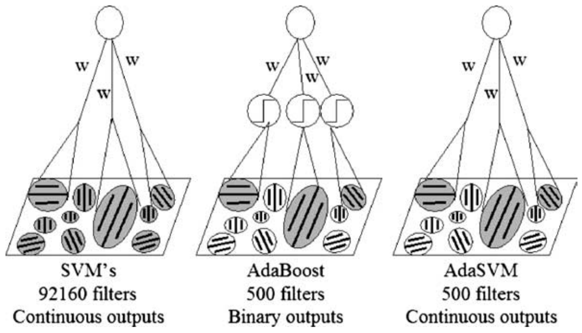

Fig. 2. SVM's learn weights for the continuous outputs of all 92,160 Gabor filters. AdaBoost selects a subset of features and learns weights for the thresholded outputs of those filters. AdaSVM's learn weights for the continuous outputs of the selected filters.

3.3.1. Distribution of spatial frequencies selected by Adaboost

The Gabor features selected by AdaBoost provide one indication of the spatial frequencies that are important for this task. Fig. 3 shows the number of chosen features at each of the five wavelengths used. Examination of this frequency distribution suggested that a wider range of spatial frequencies, particularly in the high spatial frequencies, could potentially improve performance. Indeed, by increasing from 5 to 9 spatial frequencies (2:32 pixels per cycle at 0.5 octave steps), performance of the AdaSVM improved to 93.3% correct. (See Table 2.) At this spatial frequency range, the performance advantage of AdaSVM's was greater. AdaSVM's outperformed both AdaBoost ($z=2.1$, $P=.02$) and SVM's ($z=2.6$, $p<.01$). Moreover, as the input size increases, the speed advantage of AdaSVM's becomes even more apparent. The full Gabor representation was seven times larger than before, whereas the number of Gabors selected by Adaboost only increased by a factor of 1.7. This system obtained 93.3% accuracy on a user-independent 7-alternative forced choice. Previously published results on this database were 80–88% (e.g. [3,6,29]).

We then reintroduced the approaches to multiclass SVM's found to be optimal in Section 3.1, and applied them to the AdaSVM system. Results for using all possible class partitions and training an MLR matrix instead of voting are shown in Table 4. The performance enhancement with these approaches is small, if any. Optimal performance with the AdaSVM was obtained with the simpler paradigm of 1:all partitions and voting, which is a considerable savings in training time over all possible partitions and MLR.

3.3.2. Number of support vectors

We next examined the effect of feature selection by Adaboost on the number of support vectors. Smaller numbers of support vectors proffer two advantages: (1) the classification procedure is faster, and (2) the expected generalization error decreases as the number of support vectors decreases [27]. The number of support vectors for the linear SVM ranged from 10 to 33 percent of the total number of training vectors. Nonlinear SVM's employed 14–43 percent, despite

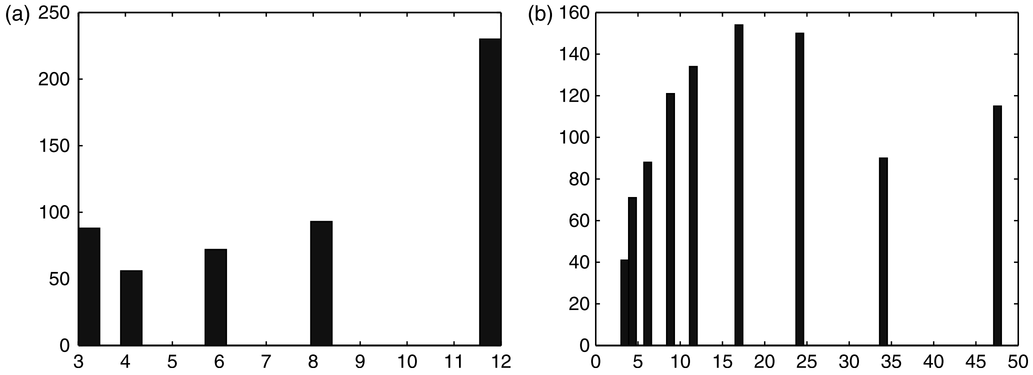

Fig. 3. Frequency distribution of features selected by Adaboost. *y*-axis is the number selected features. *x*-axis is the frequency of the Gabor filters, in cycles per face. (a) For five frequencies, ranging from 3 to 12 in half octave intervals, the distribution was skewed to the higher frequencies. (b) For nine frequencies ranging from 3 up to 48, the distribution was more balanced, peaking at roughly 17 cycles per face.

better generalization performance. Feature selection by Adaboost reduced the number of support vectors employed by the nonlinear SVM in the AdaSVM system, to 12–26 percent.

4. Comparison to linear discriminant analysis

A previous successful approach to basic emotion recognition used Linear Discriminant Analysis (LDA) to classify Gabor representations of images [20]. While LDA may be optimal when the class distributions are Gaussian, SVM's may be more effective when the class distributions are not Gaussian. Table 5 compares LDA with linear SVM's. The classifiers were tested on 48×48 images using the nine wavelength Gabor representation (2:32 pix/cyc). A small ridge term was used in LDA.

The performance results for LDA were dramatically lower than SVMs. Performance with LDA improved by adjusting the decision threshold for each emotion so as to balance the number of false detects and false negatives. This approach is labeled LDA in Tables 5 and 6. This form of threshold adjustment is commonly employed with LDA classifiers, but it uses post-hoc information, whereas the SVM performance was without post-hoc information. Even with the threshold adjustment, the linear SVM performed significantly better.

Table 4
Performance of all possible partitions and MLR for AdaSVM's. Performance is shown for nonlinear SVM's and AdaSVM's (with 900 features) for 96×96 images and 9 Gabor wavelengths (2:32)

Partitioning combining	SVM 1:all vote	AdaSVM 1: all vote	AdaSVM all poss. vote	AdaSVM all poss. MLR
	89.8	93.1	93.8	93.5

Table 5
Top row: comparing SVM performance to LDA on 48×48 pixel images

Feature selection	LDA	SVN(linear)
None	44.4	88.0
PCA	80.7	75.5
Adaboost	88.2	93.3

The two classifiers are compared with no feature selection, with feature selection by PCA, and feature selection by Adaboost.

4.1. Feature selection using PCA

Many approaches to LDA also employ PCA to perform feature selection prior to classification. For each classifier we searched for the number of PCA components which gave maximum LDA performance, which was typically 40–70 components. The PCA step resulted in a substantial improvement. The combination of PCA and threshold adjustment gave performance accuracy of 80.7% for the 7-alternative forced choice, which was comparable to other LDA results in the literature [20]. Nevertheless, the linear SVM outperformed LDA even with the combination of PCA and threshold adjustment. SVM performance on the PCA representation was significantly reduced, indicating an incompatibility between PCA and SVM's for the problem. PCA is an unsupervised feature extraction method, and may focus on aspects of the image variability that happen to be irrelevant to the task. Adaboost explicitly focusses on features relevant to the task at hand.

4.2. Feature selection using Adaboost

We next examined whether feature selection by Adaboost gave better performance with LDA than feature selection by PCA. Adaboost was used to select 900 features from 9×8×48×48=165,888 possible Gabor features which were then

Table 6
Comparison of performance with automatically located faces (top row) and hand aligned faces (lower row)

	PCA-LDA	SVM	AdaSVM
Face finder	80.7	88.0	93.3
Hand aligned	76.8	86.2	91.3

classified by LDA (Table 5). Feature selection with Adaboost gave better performance with the LDA classifier than feature selection by PCA. Using Adaboost for feature selection reduced the difference in performance between LDA and SVM's. Nevertheless, SVM's continued to outperform LDA.

4.3. Image alignment

Another difference from previous implementations of LDA for expression recognition was image alignment. Was LDA more sensitive to alignment noise than SVM's? Expression recognition performance using the automatically detected face images was compared to performance using images that were aligned using hand-labeling of internal feature points. Six points on each face image were manually located with a mouse (the corners of each eye, the nose tip, and the mouth center). Eye centers were defined as the mean of the eye corners. Images were then rotated in the plane so that the eyes were horizontal and scaled to align the eye centers as well as the midpoint between the mouth and nose tip.

As shown in Table 6, the hand alignment offered no improvement in performance over the automatically aligned face images for either LDA or SVM's. This is in part due to the fact that the images are already frontal (up to 10 degrees) and upright. This would not generalize to significant pose variations.

5. Generalization to other datasets

We tested the system on a second publicly available data set, Pictures of Facial Affect (POFA) [9]. POFA contains 110 images from 14 subjects posing facial expressions. The facial displays were guided by Ekman's observations of the facial expressions of basic emotion. The best published result on this dataset until now [5] is 90%, but this was a mean over a set of two-way forced choices. In this paper we conduct a 7-way forced choice, where chance is 14% instead of 50%.

Results are shown in Table 7. AdaSVM's trained and tested on this dataset using leave one subject out cross-validation obtained 97.3% accuracy with a linear kernel, and 95.5% with an RBF kernel. Feature selection by Adaboost had a significant impact on performance for this dataset. SVM's trained on the full set of Gabors obtained only 79.1% correct. Feature selection may be particularly important for training SVM's on smaller datasets such as this. We are currently collecting a dataset of 250 thousand images to investigate this problem further.

Table 7
Generalization performance using leave-one-out cross-validation on the POFA dataset alone and on the combined DFAT-504 and POFA datasets

	AdaSVM linear	AdaSVM RBF
POFA	97.3	95.5
DFAT-504+POFA	91.4	93.1
Train: DFAT-504 Test: POFA	56.0	60.0

The bottom row gives performance for training on DFAT-504 and testing on POFA. The AdaSVMs were tested for 96×96 images, 9 frequencies, and 953 Adaboost features.

Training and testing on a combined dataset consisting of both DFAT-504 and POFA also gave strong recognition results. Generalization performance was again tested using leave-one-subject-out cross-validation.

Generalization across datasets was substantially lower. A nonlinear AdaSVM trained on DFAT-504 and tested on POFA obtained 60% correct. This highlights the need for large training datasets of facial expressions with variations in image conditions in order to generalize across image collection environments. While the Face Finder was trained on a large number of faces (5000 positive and millions of negative examples) with many lighting conditions and other irregularities, the only condition being roughly frontal pose, the expression coder was trained on a single dataset with a uniformly controlled environment. The result is that the face finder is robust to real-world application, while the expression coder performs well only within a given dataset or combination of datasets.

6. Real-time expression recognition from video

We combined the face detection and expression recognition into a system that operates on live digital video in real time. Face detection operates at 24 frames/s in 320×240 images on a 3 ghz Pentium IV. The expression recognition step operates in less than 10 ms. Fig. 4 shows the output of the expression recognizer for a test video in which the subject posed a series of facial expressions. The traces show outputs of each of the seven emotion detectors. The output of the sadness detector increases as he poses a sad expression, and anger increases as he poses anger. The output for neutral increases as the subject passes through neutral between each expression.

Although each individual image is separately processed and classified, the outputs change smoothly as a function of

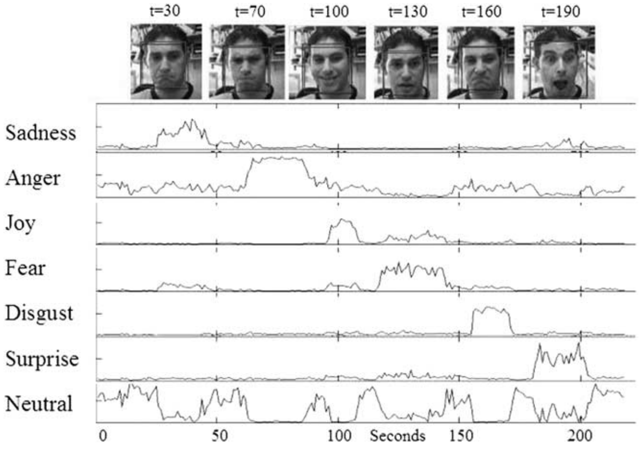

Fig. 4. Examples of real-time emotion code traces from a test video sequence. The top row shows frames from the sequence. Continuous outputs of each of the seven expression detectors is given below.

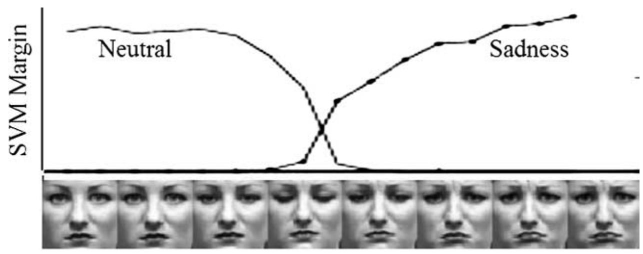

Fig. 5. Outputs of the SVM's trained for neutral and sadness for a full test image sequence of a subject performing sadness from the DFAT-504 database. The SVM output is the distance to the separating hyperplane (the margin).

time, particularly under illumination and background conditions that are favorable for alignment. (See Fig. 5). This enables applications for measuring the magnitude and dynamics of facial expressions.

To demonstrate the potential of this system we developed a real time 'emotion mirror' which renders a 3D character in real time that mimics the emotional expression of a person. (See Fig. 6). The emotion mirror is a prototype system that recognizes the emotion of the user and responds in an engaging way.

In the emotion mirror, the face-finder captures a face image which is sent to the emotion classifier. The outputs of the 7-emotion classifier constitutes a 7D emotion code. This code was sent to CU Animate, a set of software tools for rendering 3D computer animated characters in real time, developed at the Center for Spoken Language Research at CU Boulder [21]. The 7D emotion code gave a weighted combination of morph targets for each emotion.

In pilot studies for future projects, we recorded spontaneous reactions to a series of video clips and images. Fig. 7 shows the reaction of two subjects to an amusing image immediately following a distressing clip. The automatic codes for both subjects show increasing joy and decreasing disgust, but the baseline levels and the trajectories differ.

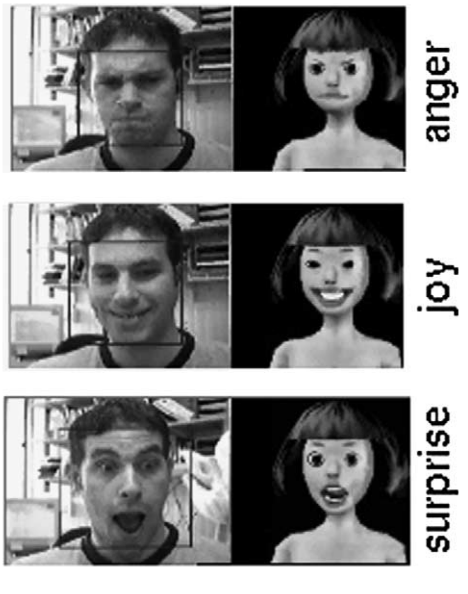

Fig. 6. Examples of the emotion mirror. The animated character mirrors the facial expression of the user.

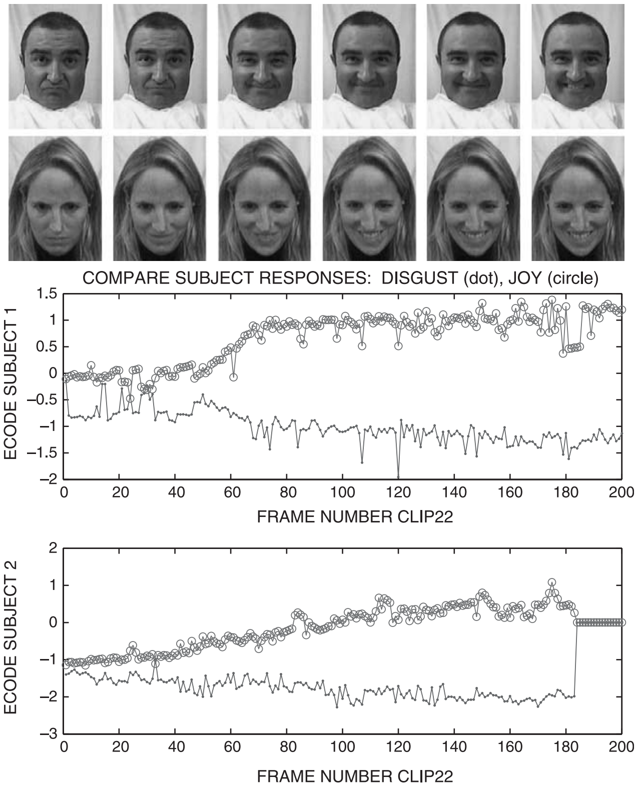

Fig. 7. Spontaneous reactions to emotive images and video. Upper section: examples of frames from video sequences of subjects 1 and 2 responding to the same stimulus. Lower section: traces for disgust (lower, blue curve) and joy (upper, green-o curve) are shown for subjects 1 and 2.

6.1. Person identification from expression dynamics

In another study we explored whether the sequence of outputs from the expression recognizer could be used for person identification. Eight subjects posed each of the six basic emotions three times over in random order. Fig. 8 shows the automatic codes for two different expressions, fear, which is difficult to pose, and surprise, which is easy to pose. We show the three trajectories for each emotion for two different subjects. The trajectories of the outputs were idiosyncratic for each person and could be used to recognize the identity of the person. Using nearest neighbor classification on the response of each of the seven emotion detectors, averaged in time windows, was sufficient to recognize the identity of the eight subjects in this pilot study with 100% accuracy.

7. Automated facial action coding

In order to objectively capture the richness and complexity of facial expressions, behavioral scientists have found it necessary to develop objective coding standards. The facial action coding system (FACS) [10] is the most objective and comprensive coding system in the behavioral sciences.

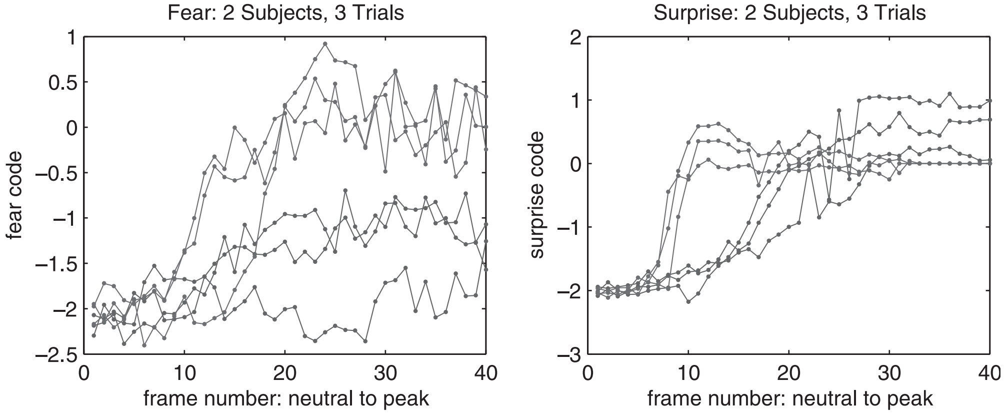

Fig. 8. Traces for repeated posed expressions of fear (left) and surprise (right) for two subjects. Subject 1 is in red/gray. Subject 2 is in blue/black.

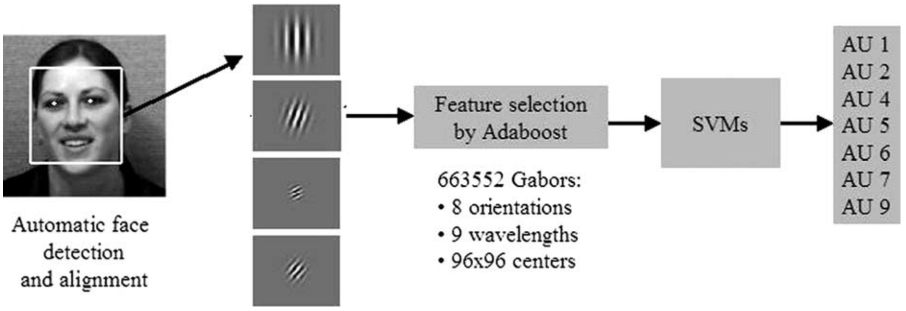

Fig. 9. Fully automated facial action coding system.

A human coder decomposes facial expressions in terms of 46 component movements. A longstanding research direction in the Machine Perception Laboratory is to automatically recognize facial actions (e.g. [1,2,8]. Three groups besides ours have focused on automatic FACS recognition as a tool for behavioral research: [23,16,26]. Systems to date still require considerable manual input, unless infrared signals are available for locating the eyes (e.g. [16]).

Here we apply the system presented above to the problem of fully automated facial action coding. The machine learning techniques presented above were repeated, where facial action labels replaced the basic emotion labels. Face images were detected and aligned automatically in the video frames and sent directly to the recognition system (Fig. 9).

The system was again trained on Cohn and Kanade's DFAT-504 dataset which contains FACS scores by two certified FACS coders in addition to the basic emotion labels. Automatic eye detection [11] was employed to align the eyes in each image. Seven support vector machines, one for each AU, were trained to detect the presence of a given AU, regardless of the co-occuring AU's. Positive examples consisted of the last (peak) frame of each sequence, and negative examples consisted of all peak frames that did not contain the target AU, plus 313 neutral images obtained from the first frame of each sequence. A nonlinear radial basis function kernel was employed. Generalization to new subjects was tested using leave-one-out cross-validation. The results are shown in Table 8. System outputs for full image sequences of test subjects are shown in Fig. 10. The subjects are from DFAT-504 and trajectories of three different action units associated with the same posed expression are shown. These are not repeated poses of the same expression.

The system obtained a mean of 92.9% agreement with human FACS labels for fully automatic recognition of seven upper facial actions. These performance rates are equal to or better than other systems tested on this dataset that employed manual registration or initialization [16,26]. The high performance rate obtained by our system is the result of many years of systematic comparisons, (such as those presented here, and also in [1,8]), investigating which image features (representations) are most effective, which classifiers are most effective, optimal resolution and spatial frequency,

Table 8
Generalization results for fully automatic recognition of 7 upper facial actions

AU	AU code	Agreement	No. examples
Inner brow raise	1	93.5	123
Outer brow raise	2	96.3	83
Brow corrugator	4	89.1	143
Upper lid raise	5	91.9	85
Cheek raise	6	93.9	93
Lower lid tight	7	87.2	85
Nose wrinkle	9	98.7	43

Agreement is measured as the percent of images correctly classified.

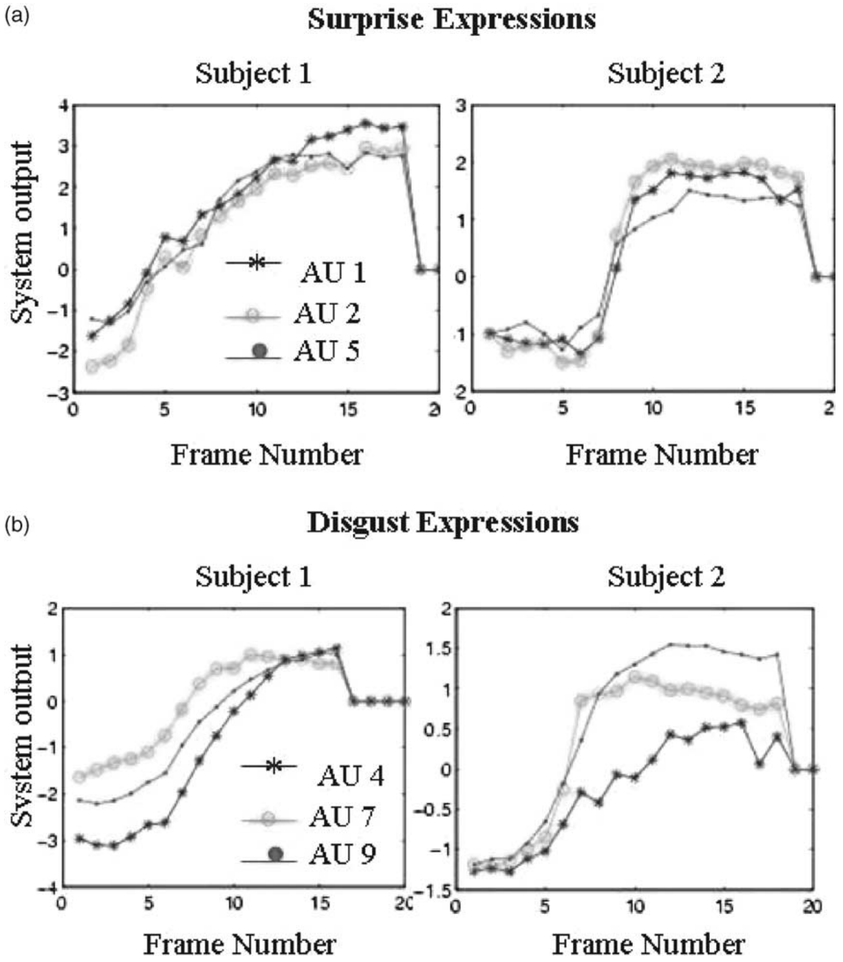

Fig. 10. Automated FACS measurements for full image sequences. (a) Surprise expression sequences from 2 subjects scored by the human coder as containing AU's 1, 2 and 5. Curves show automated system output for AU's 1, 2 and 5. (b) Disgust expression sequences from 2 subjects scored by the human coder as containing AU's 4, 7 and 9. Curves show automated system output for AU's 4, 7 and 9.

feature selection techniques, and comparing flow-based to texture-based recognition.

The approach to automatic FACS coding presented here, in addition to being fully automated, also differs from approaches such as [23,26], in that instead of designing special purpose image features for each facial action, we explore general purpose learning mechanisms for data-driven facial expression classification. These methods merge machine learning and biologically inspired models of human vision. These mechanisms can be applied to recognition of any facial action given a training data set. The approach detects not only changes in position of feature points, but also changes in image texture such as those created by wrinkles, bulges, and changes in feature shapes (Fig. 11).

8. Conclusions and future directions

We presented a systematic comparison of machine learning methods applied to the problem of fully automatic recognition of facial expressions, including AdaBoost, support vector machines, and linear discriminant analysis. We reported results on a series of experiments comparing methods for multiclass decisions, spatial frequency ranges, feature selection methods, and recognition engines. Best results were obtained by selecting a subset of Gabor filters using AdaBoost and then training Support Vector Machines on the outputs of the filters selected by AdaBoost. The combination of Adaboost and

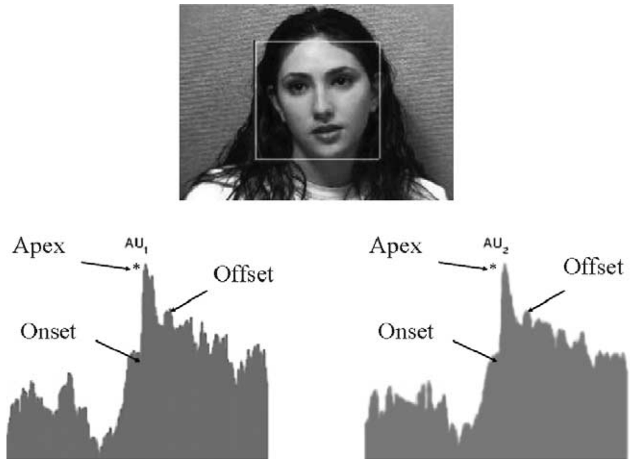

Fig. 11. Fully automated FACS detects action units 1 and 2 in a fleeting spontaneous browraise displayed in an unconstrained situation. Human coders labeled the action unit onset, apex and offset.

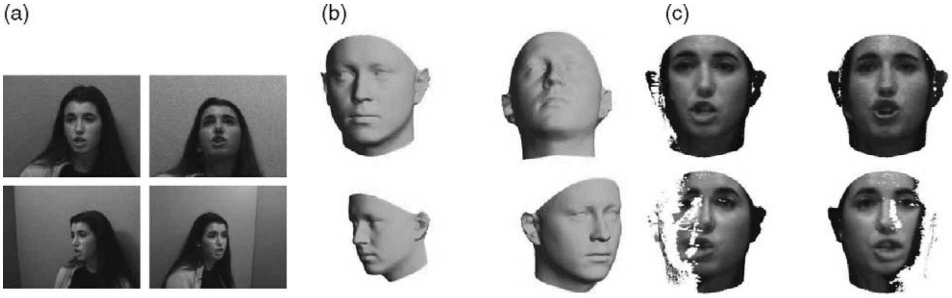

Fig. 12. Head pose estimation and warping to frontal views. (a) Four camera views of a subject at one instant. (b) Head pose estimate for each of four camera views. (c) Face images warped to frontal.

SVM's enhanced both speed and accuracy of the system. The generalization performance to new subjects for a 7-way forced choice was 93.3 and 97% correct on two publicly available datasets, the best performance reported so far on these datasets. The outputs of the classifier contain information about expression magnitude, and thus can be used to capture information about expression dynamics.

The general purpose learning mechanisms presented here for data-driven facial expression classification can be applied to recognition of any facial expression dimension given a training dataset. Here we presented results for both automatic recognition of basic emotions and automatic facial action coding.

Our results suggest that user independent, fully automatic real time coding of facial expressions in the continuous video stream is an achievable goal with present computer power, at least for applications in which frontal views can be assumed. The problem of classification of facial expressions can be solved with high accuracy by a simple linear system, after the images are preprocessed by a bank of Gabor filters. Linear systems carry a small performance penalty (92.5% instead of 93.3%) but are faster for real-time applications (see Table 3). Feature selection speeds up systems based on non-linear SVM's into the real-time range.

Our work also indicates that the current datasets may be inadequate for further progress and a new generation of dataset is greatly needed in the field. We are currently engaged in a long-term effort to develop such datasets and help accelerate progress in the field.

The automated facial expression measurement systems described above aligned faces in the 2D plane. Spontaneous behavior can contain considerable out-of-plane head rotation. The accuracy of automated facial expression measurement may be considerably improved by 3D alignment of faces. Also, information about head movement dynamics is an important component of FACS. Members of this group have developed techniques for automatically estimating 3D pose in a generative model [22] and for warping faces to frontal. See Fig. 12. In the near future, this process will be integrated into our system for recognizing expressions from video of unconstrained interactions.

We are presently exploring applications of this system including automatic evaluation of human-robot interaction [19], and deployment in automatic tutoring systems [21] and social robots. We are also exploring clinical applications, including psychiatric diagnosis and measuring response to treatment.

References

[1] Marian.S. Bartlett, Face image analysis by unsupervised learning, The Kluwer International Series on Engineering and Computer Science, vol. 612, Kluwer, Boston, Mass, 2001.

[2] M.S. Bartlett, B. Braathen, G. Littlewort-Ford, J. Hershey, I. Fasel, T. Marks, E. Smith, T.J. Sejnowski, J.R. Movellan, Automatic analysis of spontaneous facial behavior: a final project report, Technical Report UCSD MPLab TR 2001.08, University of California, San Diego, 2001.

[3] I. Cohen, N. Sebe, F. Cozman, M. Cirelo, T. Huang, Learning Baysian network classifiers for facial expression recognition using both labeled and unlabeled data, Computer Vision and Pattern Recognition (2003).

[4] N. Cristianini, J. Shawe-Taylor, Support Vector Machines, Cambridge University Press, Cambridge, 2000.

[5] M.N. Dailey, G.W. Cottrell, C. Padgett, R. Adolphs, Empath: a neural network that categorizes facial expressions, Journal of Cognitive Neuroscience 14 (8) (2002).

[6] D. Datcu, L. Rothkrantz, Automatic recognition of facial expressions using bayesian belief networks, in: Thissen, Wieringa, Pantic, Ludema (Eds.), Proceedings of the IEEE Conference on Systems Man and Cybernetics, The Hague, Netherlands, Oct 10–13, 2004.

[7] O. Dekel, Y. Singer, Multiclass learning by probabilistic embedding, in: S. Becker, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, MIT Press, Cambridge, MA, 2003.

[8] G. Donato, M. Bartlett, J. Hager, P. Ekman, T. Sejnowski, Classifying facial actions, IEEE Trans on Pattern Analysis and Machine Intelligence 21 (10) (1999) 974–989.

[9] P. Ekman, W. Friesen, Pictures of Facial Affect. Photographs, Available from Human Interaction Laboratory, UCSF, HIL-0984, San Francisco, CA 94143, 1976.

[10] P. Ekman, W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, Palo Alto, CA, 1978.

[11] I.R. Fasel, B. Fortenberry, J.R. Movellan, GBoost: a generative framework for boosting with applications to realtime eye coding. Computer Vision and Image Understanding, in press.

[12] J. Freund, R.E. Schapire, A short introduction to boosting, Journal of Japanese Society for Artificial Intelligence 14 (5) (1999) 771–780.

[13] J. Friedman, T. Hastie, R. Tibshirani, Additive Logistic Regression: A Statistical View of Boosting, 1998.

[14] S. Har-Peled, D. Roth, D. Zimak, Constraint classification for multiclass classification and ranking, in: S. Becker, K. Obermayer (Eds.), Advances in Neural Information Processing Systems, vol. 15, MIT Press, Cambridge, MA, 2003.

[15] T. Kanade, J.F. Cohn, Y. Tian, Comprehensive database for facial expression analysis, in: Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00) Grenoble, France, pp. 46-53, 2000.

[16] A. Kapoor, Y. Qi, R.W. Picard, Fully automatic upper facial action recognition, IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2003.

[17] U. Kressel, Pairwise classification and support vector machines, in: B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 255–268.

[18] M. Lades, J. Vorbrüggen, J. Buhmann, J. Lange, W. Konen, C. von der Malsburg, R. Würtz, Distortion invariant object recognition in the dynamic link architecture, IEEE Transactions on Computers 42 (3) (1993) 300–311.

[19] G. Littlewort, M.S. Bartlett, J. Chenu, I. Fasel, T. Kanda, H. Ishiguro, J.R. Movellan, Towards social robots: automatic evaluation of human-robot interaction by face detection and expression classification, in: Advances in Neural Information Processing Systems, vol. 16, MIT Press, Cambridge, MA, in press.

[20] M. Lyons, J. Budynek, A. Plante, S. Akamatsu, Classifying facial attributes using a 2-d gabor wavelet representation and discriminant analysis, Proceedings of the 4th International Conference on Automatic Face and Gesture Recognition, 2000, pp. 202–207.

[21] Jiyong Ma, Jie Yan, Ron Cole, CU animate: Tools for enabling conversations with animated characters, in: Proceedings of ICSLP-2002, Denver, USA, 2002.

[22] T.K. Marks, J. Hershey, J. Cooper Roddey, J.R. Movellan, 3d tracking of morphable objects using conditionally gaussian nonlinear filters, Computer Vision and Image Understanding, under review. See also CVPR04 Workshop: Generative-Model Based Vision.

[23] M. Pantic, J.M. Rothcrantz, Automatic analysis of facial expressions: State of the art, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1424–1445.

[24] J.C. Platt, N. Cristianini, J. Shawe-Taylor, Large margin dags for multiclass classification, in: S.A. Solla, T.K. Leen, K.-R. Muller (Eds.), Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, MA, 2000, pp. 547–553.

[25] D. Roth, M-H. Yang, N. Ahuja, Learning to recognize three dimensional objects, Neural Computation 14 (2002) 1071–1103.

[26] Y.L. Tian, T. Kanade, J.F. Cohn, Recognizing action units for facial expression analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 97–116.

[27] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, Heidelberg, DE, 1995.

[28] Paul Viola, Michael Jones, Robust Real-time Object Detection, Technical Report CRL 20001/01, Cambridge Research-Laboratory, 2001.

[29] T. Wilhelm, H. Bohme, H. Gross, A. Backhaus, Statistical and neural methods for vision based analysis of facial expressions and gender, in: Thissen, Wieringa, Pantic, Ludema (Eds.), Proceedings of the IEEE Conference on Systems Man and Cybernetics, Oct 10–13, The Hague, Netherlands, 2004.