Fully Automatic Facial Action Unit Detection and Temporal Analysis

Michel Valstar and Maja Pantic
Computing Department
Imperial College London
{M.F.Valstar, M.Pantic}@imperial.ac.uk

Abstract

In this work we report on the progress of building a system that enables fully automated fast and robust facial expression recognition from face video. We analyse subtle changes in facial expression by recognizing facial muscle action units (AUs) and analysing their temporal behavior. By detecting AUs from face video we enable the analysis of various facial communicative signals including facial expressions of emotion, attitude and mood. For an input video picturing a facial expression we detect per frame whether any of 15 different AUs is activated, whether that facial action is in the onset, apex, or offset phase, and what the total duration of the activation in question is. We base this process upon a set of spatio-temporal features calculated from tracking data for 20 facial fiducial points. To detect these 20 points of interest in the first frame of an input face video, we utilize a fully automatic, facial point localization method that uses individual feature GentleBoost templates built from Gabor wavelet features. Then, we exploit a particle filtering scheme that uses factorized likelihoods and a novel observation model that combines a rigid and a morphological model to track the facial points. The AUs displayed in the input video and their temporal segments are recognized finally by Support Vector Machines trained on a subset of most informative spatio-temporal features selected by AdaBoost. For Cohn-Kanade and MMI databases, the proposed system classifies 15 AUs occurring alone or in combination with other AUs with a mean agreement rate of 90.2% with human FACS coders.

1. Introduction

Humans interact far more naturally with each other than they do with machines. To approach the naturalness of face-to-face human interaction machines should be able to emulate the way humans communicate with each other. Although speech alone is often sufficient for communicating with another person (e.g., in a phone call), non-verbal communicative cues can help to synchronize the dialogue, to signal comprehension or disagreement and to let the dialogue run smoother, with less interruptions. With facial expressions we clarify what is said by means of lip-reading, we stress the importance of the spoken message by means of conversational signals like raising eyebrows, and we signal comprehension, disagreement, boredom and intentions [14]. Machine understanding of facial expressions could revolutionize user interfaces including ambient, automotive and robot interfaces and has become, therefore, a hot topic in AI and computer-vision research.

The method proposed in this paper intends to detect atomic facial actions called Action Units (AUs) defined by the Facial Action Coding System (FACS) [4]. FACS is the best known and the most commonly used system developed for human observers to describe facial activity in terms of visually observable facial muscle actions (AUs). Using FACS, human observers decompose a facial expression into one or more of in total 44 AUs that produced the expression in question.

Previous work on AU detection from videos includes automatic detection of 16 AUs from face image sequences using lip tracking, template matching and neural networks [15], detecting 20 AUs occurring alone or in combination by using temporal templates generated from input face video [17] and detection of 18 AUs using wavelets, AdaBoost and Support Vector Machines [1]. For a good overview of the work done on AU and emotion detection from still images or face video the reader is referred to [9, 16]. Although many of these methods are suitable for decoding the temporal segments of facial actions (onset, apex and offset), none do this explicitly [16].

The system described in this work detects 15 AUs that have a high relevance for inter-human communication. More precisely, this set of AUs is sufficient to detect the six basic emotions with high reliability [5]. These AUs are first detected for every frame of an input face video. In order to compare our results with those of other systems, we also determine which AUs have been active during the entire video. Besides a reliable detection of 15 AUs from face video we propose in this work a new method to analyze the temporal aspects of facial actions. For every AU our system detects, we determine the duration of the temporal phases onset, apex and offset.

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

To analyze a facial expression, we use 15 SVM classifiers, one for every AU we wish to detect, which are trained using features that describe the spatio-temporal relationships between 20 tracked fiducial facial points. To extract these features, we first find the face in the first frame of an input image sequence using an adapted version of the Viola and Jones face detector [19]. Within the localized face region we automatically localize 20 fiducial facial points with a facial feature point detector based on Gabor wavelets and a GentleBoost classifier [20]. After the facial points have been located in the first frame, a tracking scheme based on particle filtering with factorized likelihoods [12] is used to track the points in all subsequent frames of a video displaying a facial expression. The features are then calculated from the positions of the facial points as indicated by the point tracker. To analyse the temporal dynamics of a facial action, we use the same features as for the AU detection, only now we train a multiclass SVM for every AU to distinguish between it's neutral, onset, apex and offset phases.

The paper is organized as follows. Section 2 describes the automatic feature extraction including face detection (section 2.1), facial point localization (section 2.2), facial point tracking (section 2.3) and the final feature extraction (section 2.4). The classification schemes for AU detection and temporal analysis are described in sections 3.1 and 3.2, respectively. The datasets used in our experiments are described in section 4, while the experiments themselves are described in section 5. Finally, section 6 provides the conclusions and discusses to future directions of our research.

2. Automatic facial feature extraction

2.1. Face detection

To detect the face image in a scene we make use of a real-time face detection scheme proposed in [5], which represents an adapted version of the original Viola-Jones face detector [19]. The Viola-Jones face detector consists of a cascade of classifiers trained by AdaBoost. Each classifier employs integral image filters, which remind of Haar Basis functions and can be computed very fast at any location and scale. This is essential to the speed of the detector. For each stage in the cascade, a subset of features is chosen using a feature selection procedure based on AdaBoost. The adapted version of the Viola-Jones face detector that we employ uses GentleBoost instead of AdaBoost. It also refines the originally proposed feature selection by finding the best performing single-feature classifier from a new set of filters generated by shifting and scaling the chosen filter by two pixels in each direction, as well as composite filters made by reflecting each shifted and scaled feature horizontally about the center and superimposing it on the original. Finally the employed version of the face detector uses a smart training procedure in which, after each single feature, the system can decide whether to test another feature or to stop and make a decision. This way the system retains information about the continuous outputs of each feature detector rather than converting to binary decisions at each stage of the cascade.

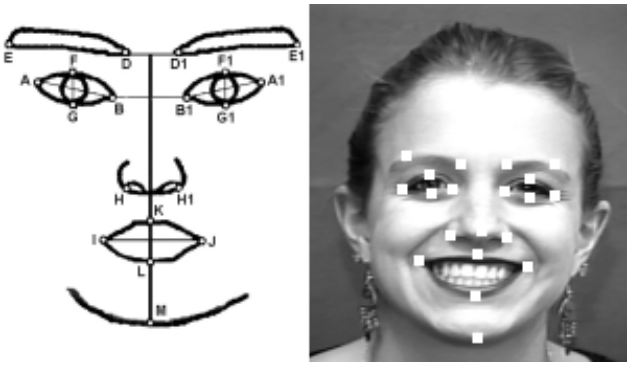

Figure 1: Fiducial facial points that will be tracked.

2.2. Facial point detection

The method that we use for fully automatic detection of 20 facial feature points (see Fig. 1) plus the irises and the center of the mouth in a face image, uses Gabor-feature-based boosted classifiers as proposed in [20]. The method (see Fig. 2) assumes that the input image is a face region, such as the output of the detection algorithm explained in section 2.1. The input face region is then divided into 20 regions of interest (ROIs), each one corresponding to one facial point to be detected. The irises and the medial point of the mouth are detected first. A combination of heuristic techniques based on the analysis of the vertical and horizontal histograms of the upper and the lower half of the face-region image achieves this. Subsequently, the detected positions of the irises and the mouth are used to localize 20 ROIs. An example of the localized ROIs for points B, I and J is depicted in Fig. 2(b).

The employed facial feature point detection method uses individual feature patch templates to detect points in the relevant ROI. These feature models are GentleBoost [6] templates build from gray level intensities and Gabor wavelet features. Recent work has shown that a Gabor approach for local feature extraction outperformed Principal Component Analysis, Fisher's Linear Discriminant and Local Feature Analysis [3]. This finding is also consistent with our experimental data that show that the vast majority of features (over 98%) selected by the utilized GentleBoost classifier were from the Gabor filter components rather than from the gray level intensities. The essence of the success of Gabor filters is that they remove most of the variability in image due to variation in lighting and contrast, while at the same

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

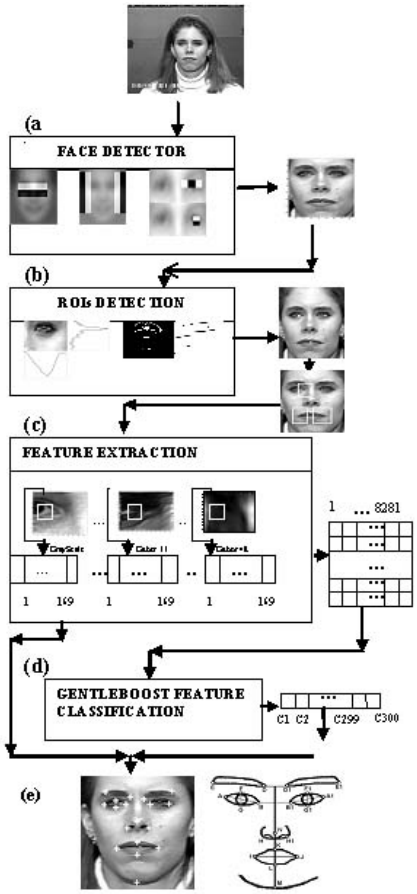

Figure 2: Outline of the fiducial facial point detection method. (a) Face detection using Haar feature based GentleBoost classifier; (b) ROI extraction, (c) feature extraction based on Gabor filtering, (d) feature selection and classification using GentleBoost classifier, (e) output of the system compared to the face drawing with facial landmark points we aim to detect

time being robust against small shift and deformation [8].

For each facial point a feature vector is extracted from the 13x13 pixels image patch centered at that point. This feature vector is used to learn the pertinent point's patch template and, in the testing stage, to predict whether the current point represents a certain facial point or not. The feature vector consists of the gray level values of the image patch and of the responses of 48 Gabor filters (8 orientations and 6 spatial frequencies, 2:12 pixels/cycle at 1/2 octave steps) taken at every pixel of the image patch. Thus, 169x49=8281 features are used to represent one point.

In the training phase, GentleBoost feature templates are learned using a representative set of positive and negative examples. As positive examples for a facial point, we used

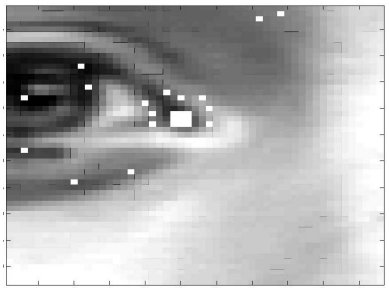

Figure 3: Positive and negative examples for training point *B*. The big white square represents the 9 positive samples. Eight negative samples have been randomly picked near the positive samples and another 8 are randomly chosen from the remainder of the region of interest.

the set of points within the 3x3 pixels region centered on the manually labeled true point. For negative training examples we used two sets of points. The first set consists of 8 points randomly displaced at a 2 pixels distance from the positive examples. The second set of negative examples consists of 8 points randomly displaced in the remaining area of the ROI (see Fig. 3). In the test phase each pixel in the ROI is filtered first by the set of 48 Gabor filters described above. Then, we use a sliding window approach in the ROI to obtain a GentleBoost response for every contained pixel, representing a measure of similarity between the 49-dimensional representation of the current test template with the learned feature point model. After scanning the entire ROI, the position with the highest response is selected as the location of the facial feature point in question.

2.3. Facial point tracking

The positions of the facial feature points in the first frame of an image sequence are automatically found using the method described in section 2.2. The positions in all subsequent frames are determined by a tracker that utilises Particle Filtering with Factorized Likelihoods (Pffl) [12]. Pffl is an extension to the Auxiliary Particle Filtering theory introduced by Pitt and Shephard [13], which in turn is an extension to classical particle filtering (Condensation) [7]. The Pffl has been initially proposed by Patras and Pantic in [12].

The main idea of particle filtering is to maintain a particle based representation of the *a posteriori* probability $p(\alpha \mid Y)$ of the state α given all the observations *Y* up to the current time instance. This means that the distribution $p(\alpha \mid Y)$ is represented by a set of pairs $\{(s_k, \pi_k)\}$ such that if s_k is chosen with probability equal to π_k, then it is as if s_k was drawn from $p(\alpha \mid Y)$. In the particle filtering framework our knowledge about the *a posteriori* probability is updated in a recursive way. Suppose that at a previous

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

time instance we have a particle based representation of the density $p(\alpha^-|Y^-)$, that is, we have a collection of K particles and their corresponding weights (i.e. $\left\{\left(s_k^-, \pi_k^-\right)\right\}$). Then, the Condensation Particle Filtering can be summarized as follows:

1. Draw K particles s_k^- from the probability density that is represented by the collection $\left\{\left(s_k^-, \pi_k^-\right)\right\}$.

2. Propagate each particle s_k^- with the transition probability $p(\alpha|\alpha^-)$ in order to arrive at a collection of K particles s_k.

3. Compute the weights π_k for each particle as follows,

$$\pi_k = p(y \mid s_k) \tag{1}$$

   Then normalize so that $\sum_k \pi_k = 1$.

This results in a collection of K particles and their corresponding weights (i.e. $\{(s_k, \pi_k)\}$ which is an approximation of the density $p(\alpha|Y)$.

The Condensation algorithm has three major drawbacks. The first drawback is that a large amount of particles that result from sampling from the proposal density $p(\alpha|Y^-)$ might be wasted because they are propagated into areas with small likelihood. The second problem is that the scheme ignores the fact that while a particle $s_k = \langle s_{k1}, s_{k2}, ..., s_{kN}\rangle$ might have low likelihood, it can easily happen that parts of it might be close to the correct solution. Finally, the third problem is that the estimation of the particle weights does not take into account the interdependencies between the different parts of the state α.

Particle filtering with factorized likelihoods [12] attempts to solve these problems in one step, given the case that the likelihood can be factorized, that is in the case that $p(y|\alpha) = \prod_i p(y|\alpha_i)$. It uses a proposal distribution $g(\alpha)$ the product of the posteriors of each α_i given the observations, that is $g(\alpha) = \prod_i p(\alpha_i|y)$, from which we draw samples s_k. These samples are then assigned weights π_k, using the same proposal distribution. We now find π_k and s_k as follows:

1. Propagate all particles s_k^- via the transition probability $p(\alpha_i|\alpha^-)$ in order to arrive at a collection of K subparticles μ_{ik}. Note, that while s_k^- has the dimensionality of the state space, the μ_{ik} have the dimensionality of the partition i.

2. Evaluate the likelihood associated with each subparticle μ_{ik}, that is let $\lambda_{ik} = p(y|\mu_{ik})$.

3. Draw K particles s_k^- from the probability density that is represented by the collection $\{(s_k^-, \lambda_{ik}\pi_k^-)\}$.

4. Propagate each particle s_k^- with the transition probability $p(\alpha_i|\alpha^-)$ in order to arrive at a collection of K sub-particles s_{ik}. Note, that s_{ik} has the dimensionality of the partition i.

5. Assign a weight π_{ik} to each sub particle as follows, $w_{ik} = \frac{p(y|s_{ik})}{\lambda_{ik}}$, $\pi_{ik} = \frac{w_{ik}}{\sum_j w_{ij}}$. With this procedure, we have a particle-based representation for each of the N posteriors $p(\alpha_i \mid y)$. That is, we have N collections $(s_{ik},)\, \pi_{ik}$, one for each i.

6. Sample K particles from the proposal function $g(\alpha) = \prod_i p(\alpha_i \mid Y)$. This is approximately equivalent to constructing each particle $s_k = \langle s_{k1}...s_{ki}...s_{kN}\rangle$ by sampling independently each s_{ik} from $p(\alpha_i \mid Y)$.

7. Assign weights π_k to the K samples as follows:

$$\pi_k = \frac{p(s_k|Y^-)}{\prod_i p(s_{ik}|Y^-)} \tag{2}$$

The weights are normalized to sum up to one. With this, we end up with a collection $\{(s_k, \pi_k)\}$ that is a particle-based representation of $p(\alpha|Y)$. Note that at the numerator of eq. 2 the interdependencies between the different sub-particles are taken into consideration. On the contrary, at the denominator, the different sub-particles are considered independent. In other words, the re-weighting process of eq. 2 favors particles for which the joint is higher than the product of the marginals.

2.4. Feature extraction

The particle filtering scheme results for every image sequence in a set of points P with dimensions $n * 20$, where n is the number of frames of the input image sequence. For all points p_i, where $i = [1:20]$ denotes the facial point, we compute first two features for every frame n:

$$\begin{aligned} f_1(\boldsymbol{p_i}) &= \boldsymbol{p_{i,y,n}} - p_{i,y,1} \\ f_2(\boldsymbol{p_i}) &= \boldsymbol{p_{i,x,n}} - p_{i,x,1} \end{aligned} \tag{3}$$

that correspond to the deviation of respectively the y and the x coordinate from the related coordinates at the first (expressionless) frame. Then, for all pairs of points $\boldsymbol{p_i}$, $\boldsymbol{p_j}$,$i \neq j$ we compute in each frame the features

$$\begin{aligned} f_3(\boldsymbol{p_i}, \boldsymbol{p_j}) &= \|\boldsymbol{p_i} - \boldsymbol{p_j}\| \\ f_4(\boldsymbol{p_i}, \boldsymbol{p_j}) &= f_3(\boldsymbol{p_i}, \boldsymbol{p_j}) - \|\boldsymbol{p_{i,1}} - \boldsymbol{p_{j,1}}\| \end{aligned} \tag{4}$$

where the norm in equation (4) is the L_2 norm. Finally, we compute the first temporal derivative df/dt of all above defined features, resulting in a set of 840 features per frame, F_n.

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

3. Facial Action Unit Analysis

3.1. Action Unit recognition

The classification of facial actions is a three-step process. First, we use a boosting algorithm to select the most important features, thus reducing the problem space and increasing the classification rates [1]. Next we use Support Vector Machines to classify the facial actions in every frame. Finally, we decide which facial actions took place across the entire image sequence by applying a dynamically learned threshold.

Boosting algorithms such as GentleBoost [6] or AdaBoost are not only fast classifiers, they are also excellent feature selection techniques. In our study, we have experimented with two different boosting techniques: GentleBoost and a simplified AdaBoost. Both algorithms use the line between the cluster centers of positive and negative samples of one feature as weak classifier. So, for 840 features (see section 2.4), we have 840 weak classifers. An advantage of feature selection by GentleBoost is that features are selected contingent on the features that have already been selected. In feature selection by GentleBoost, each feature is treated as a weak classifier. GentleBoost picks the best of those classifiers, and then boosts the weights on the examples to weight the errors more. The next feature is selected as the one that gives the best performance on the errors of the previous feature. At each step, the chosen feature can be shown to be uncorrelated with the output of the previous features. In the utilized simplified implementation of AdaBoost, we do not reweigh the distribution of the samples after each weak classification. In this way we get a simple ordering of the most important features, where the significant features may still be highly redundant. To select the final number of features to use, we apply a wrapper feature selection method. We iteratively evaluate a Support Vector Machine (SVM) with the first $k \in 1 \ldots 840$ features selected by either GentleBoost or simplified AdaBoost and choose the number of features for which the SVM performed best.

Support Vector Machines (SVMs) have proven to be very well suited for classification tasks such as facial expression recognition because, in general, the high dimensionality of the input feature space does not affect the training time, which depends only on the number of training examples. They are non-linear, generalize very well and have a well-founded mathematical basis. The essence of SVMs can be summarized in three steps: maximizing the hyperplane margin, mapping the input space to a (hopefully) linearly separable feature space and applying the 'kernel trick' to the results of the first two steps. In the remainder of this paper, α denotes the Lagrange parameters that describe the separating hyperplane in a SVM.

Maximizing the margin of the separating hyperplane $\boldsymbol{w}$ results in a high generalization ability. In words, it is the problem of finding the hyperplane that maximizes the distance between the support vectors and $\boldsymbol{w}$. This involves finding the nonzero solutions α_i of the Lagrangian dual problem, which is a quadratic programming problem and can be solved efficiently. Having found the support vector weights α_i and given a labeled training set $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ the decision function in input space is:

$$f(x) = sgn\left(\sum_{i=1}^{m} \alpha_i y_i \langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + b\right) \tag{5}$$

where b is the bias of the hyperplane, $\langle a, b \rangle$ is the inner product of a and b and m is the number of training samples. Of course, most real-world problems are not linearly separable in input space. To overcome this problem, we map each input sample $\boldsymbol{x}$ to its representation in feature space $\Phi(\boldsymbol{x})$ in which we can apply our algorithm for finding the maximal margin hyperplane. Maximizing the margin and evaluating the decision function both require the computation of the dot product $\langle \Phi(x), \Phi(x_i) \rangle$ in a high-dimensional space. These expensive calculations are reduced significantly by using a Mercer kernel K, such that

$$\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}_i) \rangle = K(\boldsymbol{x}, \boldsymbol{x}_i) \tag{6}$$

The patterns which we want to detect using our maximal margin classifier do not need to coincide with the input $\boldsymbol{x}$. We may as well apply our decision function (5) directly on $\Phi(x)$. Substituting (6) for the inner product, the decision function in feature space directly becomes

$$f(\boldsymbol{x}) = sgn\left(\sum_{i=1}^{m} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b\right) \tag{7}$$

To detect 15 different AUs occurring alone or in combination in an input image sequence, we used 15 separate SVMs to perform binary decision tasks using one-versus-all partitioning of the data resulting from the feature extraction and selection stages described above.

3.2. Facial action dynamics

A facial action, in our case an AU activation, can be in any one of four possible phases: (i) the onset phase, where the muscles are contracting and the appearance of the face changes as the facial action grows stronger, (ii) the apex phase, where the facial action is at its peak and there are no more changes in facial appearance due to this particular facial action, (iii) the offset phase, where the muscles are relaxing and the face returns to its neutral appearance and (iv) the neutral phase, where there are no signs of activation of this particular facial action. Often the order of these phases is neutral-onset-apex-offset-neutral, but other combinations

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

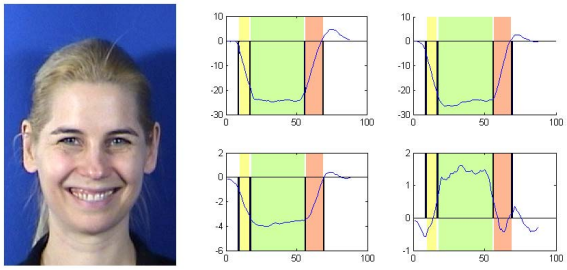

Figure 4: Temporal analysis of Action Unit 12 (smile). Shown are the four most informative features for AU12. The yellow shaded area depicts the period manually labeled as onset, the green area depicts the apex phase and the orange area depicts the offset phase.

such as multiple-apex facial actions are possible as well. As every facial action can be divided into these four temporal segments, we consider the problem to be a four-valued multiclass classification problem .

We used a one-vs-one approach to multiclass SVMs (mc-SVMs). In this approach, we train a separate specialized sub-classifier for every combination of classes, resulting in this case in $\sum_{i=1}^{C-1} i = 6$ sub-classifiers ($C = 4$ being the number of classes in our multiclass problem). When a new test sample is introduced to the mc-SVM, every sub-classifier returns a prediction of the class, and a majority vote is cast to determine the final output of the mc-SVM. Again, we first use boosting feature selection techniques to determine which features will be used for training and testing the sub-classifiers.

4. Utilized dataset

For the AU detection study we have used data from the commonly used Cohn-Kanade facial expression database [4]. This database consists of gray scale recordings of subjects displaying six basic expressions of emotion on command. The part of the database that is available from the authors upon request consists of a total of 487 gray scale recordings of 97 subjects. From a total of 45 AUs, 15 AUs can be recognized based upon the motion of the 20 facial feature points (see Table. 1). The subset of the database that we use for our AU detection validation study considers the 15 AUs in question and consists of 153 image sequences of 66 subjects. These image sequences were subsequently AU coded frame by frame by two experts, based on the available AU event coding that is provided together with the Cohn Kanade database.

One drawback of the available Cohn-Kanade database, is that the image sequences stop once the facial expression shown reaches its apex phase. Therefore it is not suitable for temporal analysis. To evaluate the proposed method for AU temporal analysis, we use the MMI-Facial Expression Database [11] for the temporal analysis. The pertinent database contains over 800 face video sequences recorded in true color, starting and ending in the neutral phase with a full temporal onset-apex-offset pattern in between. The database has been developed as a web-based direct-manipulation application, allowing easy access and easy search of the available images.

5. Experiments

5.1. Action Unit Detection

For the training of our fully automatic AU detector, we use features that result from tracking manually initialized facial points. We then test the trained system using features resulting from tracking the automatically localized feature points. The performance for every AU is evaluated separately using the previously mentioned 153 image sequences of 66 subjects from the Cohn-Kanade database, using a 66-fold person independent leave-one-subject-out cross validation (cv) scheme. To avoid over-fitting to our training data and thus increase the generalization performance of our system, we employed within each fold of this 66-fold outer cv loop a three fold inner cv-loop to obtain the optimal SVM parameters and select the optimal features using GentleBoost. In this inner cv loop we randomly split the training data of the outer loop in two, one part to train the GentleBoost and SVM classifiers, and the second part to test the trained classifiers on. Again, to increase generalization we split the dataset per image sequence, so that all samples from one image sequence are either in the training set or in the test set. This process is repeated three times to achieve more stable results for the parameter values and feature selection. We train our SVMs using a radial basis frequency kernel $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\right)$. Thus, the three parameters to optimize are the kernel width σ, the penalty parameter C, and the number of boosting selected features to include. All features have been normalized to have zero mean and a standard deviation of one.

We evaluated two different boosting techniques as feature selectors; simplified AdaBoost and GentleBoost. Table 1 part A shows the results of the AU recognition study using the simplified AdaBoost while part B shows the results using GentleBoost. As the results clearly show, the simplified AdaBoost outperforms GentleBoost as a feature selector when used on our data. The results for our fully automatic AU detection method are encouraging. For instance, Valstar et al reported an average classification rate of 95% using a similar method [18], but while using manual initialization of various facial points.

Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)
0-7695-2646-2/06 $20.00 © 2006 IEEE

Table 1: Action Unit recognition results using simplified AdaBoost (part A) and GentleBoost (part B) selected features and Support Vector Machine classifiers

	A			B		
AU	AU	rec	pre	AU	rec	pre
1	0.915	0.821	0.939	0.869	0.768	0.860
2	0.967	0.929	0.951	0.954	0.929	0.907
4	0.889	0.868	0.881	0.869	0.809	0.887
5	0.902	0.818	0.750	0.882	0.697	0.742
6	0.941	1.00	0.786	0.941	0.970	0.800
7	0.765	0.779	0.759	0.784	0.831	0.762
9	0.915	0.840	0.700	0.830	0.520	0.481
10	0.791	0.410	0.640	0.797	0.410	0.667
12	0.961	0.972	0.875	0.928	0.972	0.778
15	0.908	0.214	0.500	0.882	0.158	0.432
20	0.869	0.556	0.652	0.882	0.556	0.714
24	0.922	0.077	1.00	0.882	0.077	0.143
25	0.941	0.979	0.929	0.863	0.915	0.869
26	0.876	0.656	0.865	0.876	0.674	0.886
27	0.961	1.00	0.806	0.954	0.920	0.821
Average:	0.902	0.728	0.802	0.821	0.680	0.717

Table 2: Quantitative results of temporal analysis of facial actions, per Action Unit. From left to right: AU, fraction of facial action patterns found, relative duration of facial action, time shift of facial action in frames and classification rates of the onset, apex and offset phases.

AU	Correct pat.	rel. dur.	shift	onset	apex	offset
1	0.941	0.913	3.13	0.907	0.901	0.907
2	1.00	0.912	4.86	0.844	0.785	0.892
4	0.758	1.10	9.20	0.899	0.742	0.912
5	1.00	0.835	11.2	0.815	0.669	0.875
6	0.938	1.63	8.80	0.947	0.863	0.914
7	0.857	1.64	9.96	0.948	0.749	0.952
9	1.00	0.969	3.00	0.935	0.915	0.931
10	1.00	0.794	3.41	0.919	0.854	0.890
12	1.00	1.02	4.73	0.920	0.841	0.883
15	0.917	0.85	10.9	0.910	0.726	0.929
20	1.00	0.976	4.55	0.906	0.887	0.905
24	1.00	1.22	13.0	0.899	0.665	0.931
25	0.886	0.868	6.20	0.912	0.863	0.887
26	0.905	0.982	3.00	0.931	0.871	0.879
27	1.00	0.980	4.63	0.925	0.944	0.946

5.2. Temporal Analysis

To evaluate the method for discerning the four temporal segments neutral, onset, apex and offset, we used 171 samples from the MMI facial expression database and evaluated 15 mc-SVMs, one for every Action Unit we can detect using our AU detection method. Again we use a leave-one-person-out outer cv-loop and a three-fold inner cv loop for optimal generalization and person independence of our system. The mc-SVMs are trained using manually initialized tracking data and tested using the automatically initialized tracking data. We only use image sequences that contain the AU that the mc-SVM learns to analyze, as we intend to apply this temporal analysis as a second stage after the AU detector predicted that AU to be present. We adopted AdaBoost as the feature selector, as the results from the AU detection experiments clearly showed that for this type of data AdaBoost outperforms GentleBoost. Besides the classification rates for the onset, apex and offset phases, we also measured the predicted duration of a facial action relative to the actual duration, the error in the prediction of the beginning of the facial action (in frames) and how often a facial action pattern was properly detected overall. These results are presented in table 2. They suggest that the proposed method achieves an excellent detection of the temporal patterns. On average, 95.0% of the temporal patterns were detected correctly. The duration of most AUs was analysed well too. Only for AU6 and AU7 did the system perform bad, with a measurement of the duration that was over 60% off from the actual duration of those facial actions. It seems that human observers detect activation of these AUs not only based on the presence of a certain movement (e.g. an upward movement of the lower eyelid), but also based on the appearance of the facial region around the eye corner (e.g. crow feet wrinkles in the case of AU6). As such an appearance lasts shorter than the movement of the lower eyelid, the actual duration is much shorter than the predicted duration of the activation.

6. Conclusion

In this paper we extended the work on automatic facial expression analysis from face video with two new key features that are essential to achieve our ultimate goal, that is, fully automated fast and robust facial expression analysis from face video. The first feature is the method for automatic localization of 20 facial feature points. With this method we have an automatic initialisation of our facial feature tracker. The AU detection results suggest that this method works well compared to similar methods employing manually initialization of facial features. The second novel feature is the detection of temporal segments of facial actions. In almost all cases we find the correct temporal pattern. The duration of the predicted temporal behavior of a facial action is close to the actual duration, except for the AU6 and AU7 which have a predicted duration of over

1.6 times the actual duration, on average. As explained in section 5 and since appearance based analysis is not performed by our system, only the movement of the points determine the duration of the AU activation, which usually has a longer duration than the relevant change in the appearance of the eye corner, causing the predicted AU duration to be longer than it actually is.

At this point we can only detect and track the facial feature points in near-frontal view imagery. In the future, we wish to extend this to orientation independent facial feature point detection and tracking. Another issue will be to extend this method so that we are able to detect more Action Units.

Acknowledgements

The authors would like to thank Jeffrey Cohn of the University of Pittsburgh for providing the Cohn-Kanade database. The work of M.F. Valstar has been supported by the Netherlands BSIK-MultimediaN-N2 Interaction project. The work of M. Pantic has been supported by the Netherlands Organization for Scientific Research (NWO) Grant EW-639.021.202. The work has been conducted while the authors were at Delft University of Technology.

References

[1] M.S. Bartlett, G. Littlewort,C. Lainscsek, I. Fasel, J. Movellan, "Machine Learning Methods for Fully Automatic Recognition of Facial Expressions and Actions", *Proc. IEEE Int'l Conf. Systems Man and Cybernetics,* vol 1, pp. 592-597, 2004

[2] J.F. Cohn and K.L. Schmidt, "The timing of facial motion in posed and spontaneous smiles", *Int'l J. Wavelets, Multiresolution and Information Processing,* vol. 2, pp 1-12, 2004

[3] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, T.J. Sejnowski, "Classifying Facial Actions", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, No. 10, pp. 974-989, 1999

[4] P. Ekman, W.V. Friesen and J.C. Hager, "The Facial Action Coding System: A Technique for the Measurement of Facial Movement", San Francisco: Consulting Psychologist, 2002

[5] I.R. Fasel, B. Fortenberry and J.R. Movellan, "A generative framework for real time object detection and classification", *Int'l J Computer Vision and Image Understanding*, vol. 98, no. 1, pp. 181-210, 2005

[6] J. Friedman, T. Hastie, and R. Tibshirani. "Additive logistic regression: a statistical view of boosting", *The Annals of Statistics*, 28(2), pp. 337-374, April 2000

[7] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking", *Int'l J. Computer Vision*, pp. 5-28, 1998

[8] M. Osadchy, D. W. Jacobs and M. Lindenbaum, "Surface Dependent Representations for Illumination Insensitive Image Comparison" *Technion CIS,* vol. 2, 2005

[9] M. Pantic and L.J.M. Rothkrantz, "Toward an Affect-Sensitive Multimodal Human-computer Interaction", *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1370-1390, 2003

[10] M. Pantic and L.J.M. Rothkrantz, "Facial action recognition for facial expression analysis from static face images", *IEEE trans. on Systems, Man and Cybernetics Part B*, vol. 34, pp. 1449-1461, 2004

[11] M.Pantic, M.F. Valstar, R. Rademaker and L. Maat, "Web-based database for facial expression analysis", *In ICME'05*, pp. 317-321, 2005

[12] I. Patras and M. Pantic, "Particle filtering with factorized likelihoods for tracking facial features", *In FG'04,* pp. 97-102, 2004

[13] M.K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filtering. *J. American Statistical Association,* vol. 94, pp. 590-599, 1999

[14] J.A. Russell and J.M. Fernandez-Dols, Eds., *The Pshychology of Facial Expression*, New York: Cambridge University Press, 1997

[15] Y. Tian, T. Kanade and J.F. Cohn, "Recognizing action units for facial expression analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97-115, 2001

[16] Y.L. Tian, T. Kanade and J.F. Cohn, "Facial Expression Analysis", in: S.Z. LI, A.K. Jain (eds.): Handbook of Face Recognition, Springer, New York, 2005

[17] M.F. Valstar, I. Patras and M. Pantic, "Motion history for facial action detection in video", *Proc. IEEE Int'l Conf. Systems Man and Cybernetics,* vol 1, pp. 635-640, 2004

[18] M.F. Valstar, I. Patras and M. Pantic, "Facial Action Unit Detection using Probabilistic Actively Learned Support Vector Machines on Tracked Facial Point Data", *In CVPR'05,* vol.3, pp. 76-84, 2005

[19] P. Viola and M. Jones, "Robust real-time object detection", Technical Report CRL 20001/01, Cambridge Research Laboratory, 2001

[20] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using gabor feature based boosted classifiers", *In SMC'05,* pp. 1692-1698, 2005