Dynamic Cascades with Bidirectional Bootstrapping for Action Unit Detection in Spontaneous Facial Behavior

Yunfeng Zhu, Fernando De la Torre, Jeffrey F. Cohn, *Associate Member, IEEE,* and Yu-Jin Zhang, *Senior Member, IEEE*

Abstract—Automatic facial action unit detection from video is a long standing problem in facial expression analysis. Research has focused on registration, choice of features, and classifiers. A relatively neglected problem is the choice of training images. Nearly all previous work uses one or the other of two standard approaches. One approach assigns peak frames to the positive class and frames associated with other actions to the negative class. This approach maximizes differences between positive and negative classes, but results in a large imbalance between them, especially for infrequent AUs. The other approach reduces imbalance in class membership by including all target frames from onsets to offsets in the positive class. However, because frames near onsets and offsets often differ little from those that precede them, this approach can dramatically increase false positives. We propose a novel alternative, dynamic cascades with bidirectional bootstrapping (DCBB) to select training samples. Using an iterative approach, DCBB optimally selects positive and negative samples in the training data. Using Cascade Adaboost as basic classifier, DCBB exploits the advantages of feature selection, efficiency, and robustness of Cascade Adaboost. To provide a real-world test, we used the RU-FACS (a.k.a. M3) database of non-posed behavior recorded during interviews. For most tested action units, DCBB improved AU detection relative to alternative approaches.

Index Terms—Facial expression analysis, action unit detection, FACS, dynamic cascade boosting, bidirectional bootstrapping.

1 Introduction

The face is one of the most powerful channels of nonverbal communication. Facial expression provides cues about emotional response, regulates interpersonal behavior, and communicates aspects of psychopathology. To make use of the information afforded by facial expression, Ekman and Friesen [1] proposed the Facial Action Coding System (FACS). FACS segments the visible effects of facial muscle activation into "action units (AUs)". Each action unit is related to one or more facial muscles. These anatomic units may be combined to represent more molar facial expressions. Emotion-specified joy, for instance, is represented by the combination of AU6 (cheek raiser, which results from contraction of the orbicularis occuli muscle) and AU12 (lip-corner puller, which results from contraction of the zygomatic major muscle). The FACS taxonomy was developed by manually observing live and recorded facial behavior and by recording the electrical activity of underlying facial muscles [2]. Because of its descriptive power, FACS has become the state of the art in manual measurement of facial expression [3] and is widely used in studies of spontaneous facial behavior [4]. For these and related reasons, much effort in automatic facial image analysis seeks to automatically recognize FACS action units [5–9].

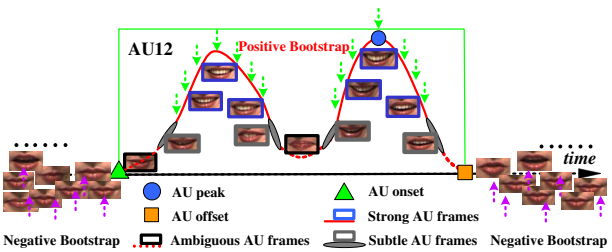


Fig. 1. Example of strong, subtle, and ambiguous samples of FACS action unit 12. Strong samples typically correspond to the peak, or maximum intensity, and ambiguous frames correspond to AU onset and offset and frames proximal to them. Subtle samples are located between strong and ambiguous ones. Using an iterative algorithm, DCBB selects positive samples such that the detection accuracy is optimized during training.

Automatic detection of AUs from video is a challenging problem for several reasons. Non-frontal pose and moderate to large head motion make facial image registration difficult, large variability occurs in the temporal scale of

This work was supported in part National Institute of Health Grant 51435. *The first author was also partially supported by a scholarship from China Scholarship Council.*
Yunfeng Zhu is in the Department of Electronic Engineering at Tsinghua University, Beijing 100084 China. mail: zhu-yf06@mails.tsinghua.edu.cn
Fernando De la Torre, is in the Robotics Institute at Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 USA. mail: ftorre@cs.cmu.edu
Jeffrey F. Cohn is in the Department of Psychology at University of Pittsburgh, Pittsburgh, Pennsylvania 15260 USA. mail: jeffcohn@pitt.edu
Yu-Jin Zhang is in the Department of Electronic Engineering at Tsinghua University, Beijing 100084 China. mail: zhang-yj@mail.tsinghua.edu.cn

facial gestures, individual differences occur in shape and appearance of facial features, many facial actions are inherently subtle, and the number of possible combinations of 40+ individual action units numbers in the thousands. More than 7000 action unit combinations have been observed [4]. Previous efforts at AU detection have emphasized types of features and classifiers. Features have included shape and various appearance features, such as grayscale pixels, edges, and appearance (e.g., canonical appearance, Gabor, and SIFT descriptors). Classifiers have included support vector machines (SVM) [10], boosting [11], hidden Markov models (HMM) [12] and dynamic Bayesian networks (DBN) [13] review much of this literature.

By contrast, little attention has been paid to the assignment of video frames to positive and negative classes. Typically, assignment has been done in one of two ways. One assigns to the positive class those frames at the peak of each AU or proximal to it. Peak refers to frame of maximum intensity of an action unit between when it begins ("onset") and when it ends ("offset"). The negative class then is chosen by randomly sampling other AUs, including AU 0 or neutral. This approach suffers at least two drawbacks: (1) the number of training examples will often be small, which results in a large imbalance between positive and negative frames; and (2) peak frames may provide too little variability to achieve good generalization. These problems may be circumvented by following an alternative approach; that is to include all frames from onset to offset in the positive class. This approach improves the ratio of positive to negative frames and increases representativeness of positive examples. The downside is confusability of positive and negative classes. Onset and offset frames and many of those proximal or even further from them may be indistinguisable from the negative class. As a consequence, the number of false positives can dramatically increase.

To address these issues, we propose an extension of Adaboost [14–16] called Dynamic Cascades with Bidirectional Bootstrapping (DCBB). Fig. 1 illustrates the main idea. Having manually annotated FACS data with onset, peak, and offset, the question we address is how best to select the AU frames for the positive and negative class. Preliminary results for this work has been presented in [17].

In contrast to previous approaches to class assignment, DCBB automatically distinguishes between strong, subtle, and ambiguous frames for AU events of different intensity. Strong frames correspond to the peaks and the ones proximal to them; ambiguous frames are proximal to onsets and offsets; subtle frames occur between strong and ambiguous ones. Strong and subtle frames are assigned to the positive class. By distinguishing between these three types, DCBB maximizes the number of positive frames while reducing confusability between positive and negative classes.

For high intensity AUs in comparison with low intensity AUs, the algorithm will select more frames for the positive class. Some of these frames may be similar in intensity to low intensity AUs. Similarly, if multiple peaks occur between an onset and offset, DCBB assigns multiple segments to the positive class. See Fig. 1 for an example. Strong and subtle but not ambiguous AU frames are assigned to the positive class. For the negative class, DCBB proposes a mechanism, which is similar as Cascade AdaBoost to optimize that as well, the principles are that the weight of misclassified negative class will be increased during training step of each weak classifier, and don't learning to much at each cascade stage. Moreover, the positive class is changed at each iteration, while the corresponding negative class is reselected again.

In experiments, we evaluated the validity of our approach to class assignment and selection of features. In the first experiment, we illustrate the importance of selecting the right positive samples for action unit detection. In the second we compare DCBB with standard approaches based on SVM and AdaBoost.

The rest of the paper is organized as follows. Section 2 reviews previous work on automatic methods for action unit detection. Section 3 describes pre-processing steps for alignment and feature extraction. Section 4 gives details of our proposed DCBB method. Section 5 provides experimental results in non-posed, naturalistic video. For experimental evaluation, we used FACS-coded interviews from the RU-FACS (a.k.a. M3) database [18, 19]. For most action units tested, DCBB outperformed alternative approaches.

2 Previous Work

This section describes previous work on FACS and on automatic detection of AUs from video.

2.1 FACS

The Facial Action Coding System (FACS) [1] is a comprehensive, anatomically-based system for measuring nearly all visually discernible facial movement. FACS describes facial activity on the basis of 44 unique action units(AUs), as well as several categories of head and eye positions and movements. Facial movement is thus described in terms of constituent components, or AUs. Any facial expression may be represented as a single AU or a combination of AUs. For example, the felt, or Duchenne, smile is indicated by movement of the zygomatic major (AU12) and orbicularis oculi, pars lateralis (AU6). FACS is recognized as the most comprehensive and objective means for measuring facial movement currently available, and it has become the standard for facial measurement in behavioral research in psychology and related fields. FACS coding procedures allow for coding of the intensity of each facial action on a 5-point intensity scale, which provides a metric for the degree of muscular contraction and for measurement of the timing of facial actions. FACS scoring produces a list of AU-based descriptions of each facial event in a video record. Fig. 2 shows an example for AU12.

2.2 Automatic FACS detection from video

Two main streams in the current research on automatic analysis of facial expressions consider emotion-specified

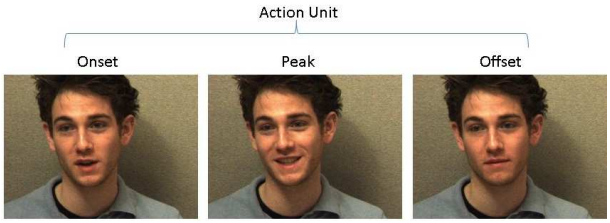


Fig. 2. FACS coding typically involves frame-by-frame inspection of the video, paying close attention to transient cues such as wrinkles, bulges, and furrows to determine which facial action units have occurred and their intensity. Full labeling requires marking onset, peak and offset and may include annotating changes in intensity as well. Left to right, evolution of an AU 12 (involved in smiling), from onset, peak, to offset.

expressions (e.g., happy or sad) and anatomically based facial actions (e.g., FACS). The pioneering work of Black and Yacoob [20] recognizes facial expressions by fitting local parametric motion models to regions of the face and then feeding the resulting parameters to a nearest neighbor classifier for expression recognition. De la Torre et al. [21] used condensation and appearance models to simultaneously track and recognize facial expression. Chang et al. [22] learned a low dimensional Lipschitz embedding to build a manifold of shape variation across several people and then used I-condensation to simultaneously track and recognize expressions. Lee and Elgammal [23] employed multi-linear models to construct a non-linear manifold that factorizes identity from expression.

Several promising prototype systems were reported that can recognize deliberately produced AUs in either near frontal view face images (Bartlett et al., [24]; Tian et al., [8]; Pantic & Rothkrantz, [25]) or profile view face images (Pantic & Patras, [26]). Although high scores have been achieved on posed facial action behavior [13, 27, 28], accuracy tends to be lower in the few studies that have tested classifiers on non-posed facial behavior [11, 29, 30]. In non-posed facial behavior, non-frontal views and rigid head motion are common, and action units are often less intense, have different timing, and occur in complex combinations [31]. These factors have been found to reduce AU detection accuracy [32]. Non-posed facial behavior is more representative of facial actions that occur in real life, which is our focus in the current paper.

Most work in automatic analysis of facial expressions differs in the choice of facial features, representations, and classifiers. Barlett et al. [11, 19, 24] used SVM and AdaBoost in texture-based image representations to recognize 20 action units in near-frontal posed and non-posed facial behavior. Valstar and Pantic [26, 33, 34] proposed a system that enables fully automated robust facial expression recognition and temporal segmentation of onset, peak, and offset from video of mostly frontal faces. The system included particle filtering to track facial features, Gabor-based representations, and a combination of SVM and HMM to temporally segment and recognize action units. Lucey et al. [30, 35] compared the use of different shape and appearance representations and different registration mechanisms for AU detection.

Fig. 3. AAM tracking across several frames

Tong et al. [13] used Dynamic Bayesian Networks with appearance features to detect facial action units in posed facial behavior. The correlation among action units served as priors in action unit detection. Comprehensive reviews of automatic facial coding may be found in [5–8, 36].

To the best of our knowledge, no previous work has considered strategies for selecting training samples or evaluated their importance in AU detection. This is the first paper to propose an approach to optimize the selection of positive and negative training samples. Our findings suggest that a principled approach to optimizing the selection of training samples increases accuracy of AU detection relative to current state of the art.

3 Facial Feature Tracking and Image Features

This section describes the system for facial feature tracking using active appearance models (AAMs), and extraction and representation of shape and appearance features for input to the classifiers.

3.1 Facial tracking and alignment

Over the last decade, appearance models have become increasingly important in computer vision and graphics. Parameterized Appearance Models (PAMs) (e.g. Active Appearance Models [37–39] and Morphable Models [40]) have been proven useful for detection, facial feature alignment, and face synthesis. In particular, Active Appearance Models (AAMs) have proven an excellent tool for aligning facial features with respect to a shape and appearance model. In our case, the AAM is composed of 66 landmarks that deform to fit perturbations in facial features. Person-specific models were trained on approximately 5% of the video [39]. Fig. 3 shows an example of AAM tracking of facial features in a single subject from the RU-FACS [18, 19] video data-set.

After tracking facial features using AAM, a similarity transform registers facial features with respect to an average face (see middle column in Fig. 4). In the experiments reported here, the face was normalized to 212×212 pixels.

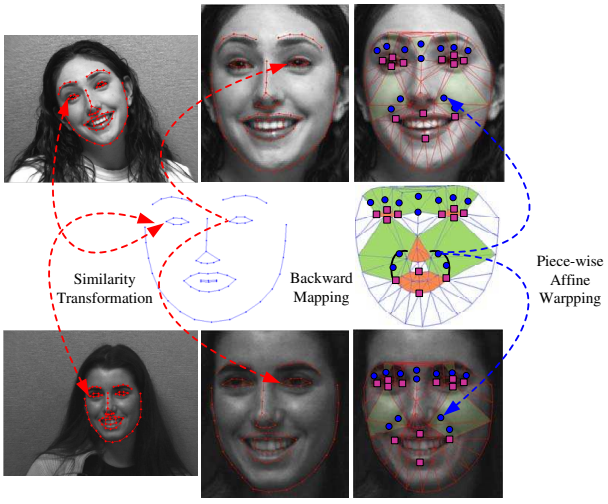


Fig. 4. Two-step alignment

To extract appearance representations in areas that have not been explicitly tracked (e.g. nasolabial furrow), we use a backward piece-wise affine warp with Delaunay triangulation to set up the correspondence. Fig. 4 shows the two step process for registering the face to a canonical pose for AU detection. Purple squares represent tracked points and blue dots represent meaningful non-tracked points. The dashed blue line shows the mapping between a point in the mean shape and its corresponding point in the original image. This two-step registration proves important toward detecting low intensity action units.

3.2 Appearance features

Appearance features for AU detection [11, 41] outperformed shape only features for some action units; see Lucey et al. [35, 42, 43] for a comparison. In this section, we explore the use of the SIFT [44] descriptors as appearance features.

Given feature points tracked with AAMs, SIFT descriptors are first computed around the points of interest. SIFT descriptors are computed from the gradient vector for each pixel in the neighborhood to build a normalized histogram of gradient directions. For each pixel within a subregion, SIFT descriptors add the pixel's gradient vector to a histogram of gradient directions by quantizing each orientation to one of 8 directions and weighting the contribution of each vector by its magnitude.

4 Dynamic Cascades with Bidirectional Bootstrapping (DCBB)

This section explores the use of a dynamic boosting techniques to select the positive and negative samples that improve detection performance in AU detection.

Bootstrapping [45] is a resampling method that is compatible with many learning algorithms. During the bootstrapping process, the active sets of negative examples are extended by examples that were misclassified by the current classifier. In this section, we propose Bidirectional Bootstrapping, a method to bootstrap both positive and negative samples.

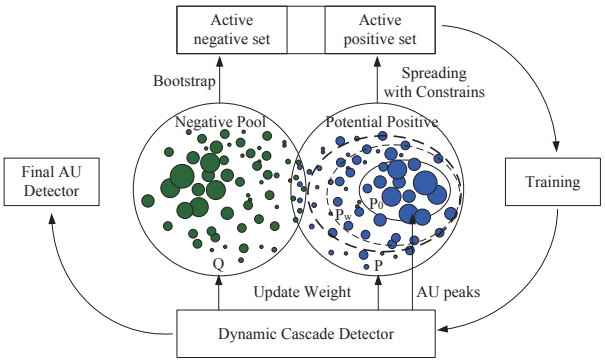


Fig. 5. Bidirectional Bootstrapping.

Bidirectional Bootstrapping beings by selecting as positive samples only the peak frames and uses Classification and Regression Tree (CART) [46] as a weak classifier (Initial learning in Section 4.1). After this initial training step, Bidirectional Bootstrapping extends the positive samples from the peak frames to proximal frames and redefines new provisional positive and negative training sets (Dynamic learning in Section 4.2). The positive set is extended by including samples that are classified correctly by the previous strong classifier (Cascade AdaBoost in our algorithm), the negative set is extended by examples misclassified by the same strong classifier, thus emphasizing negative samples close to the decision boundary. With the bootstrapping of positive samples, the generalization ability of the classifier is gradually enhanced. The active positive and negative sets then are used as an input to the CART that returns a hypothesis, which updates the weights in the manner of Gentle AdaBoost [47], and the training continues until the variation between previous and current Cascade AdaBoost become smaller than a defined threshold. Fig. 5 illustrates the process. In Fig. 5 P is potential positive data set, Q is negative data set (Negative Pool), P_0 is the positive set in Initial learning step, P_w is the active positive set in each iteration, the size of solid circle illustrate the intensity of AU samples, the right ellipses illustrate the spreading of dynamic positive set. See details in Algorithm 1 and 2.

4.1 Initial training step

This section explains the initial training step for DCBB. In the initial training step we select the peaks and the two neighboring samples as positive samples, and a randomly selected sample of other AUs and non-AUs as negative samples. As in standard AdaBoost [14], we define the false positive target ratio (F_r), the maximum acceptable false positive ratio per cascade stage (f_r), and the minimum acceptable true positive ratio per cascade stage (d_r). The initial training step applies standard AdaBoost using CART [46] as a weak classifier as summarized in Algorithm 1.

Input:

- Positive data set P_0 (contains AU peak frames p and $p \pm 1$);
- Negative data set Q (contains other AUs and non-AUs);
- Target false positive ratio F_r;
- Maximum acceptable false positive ratio per cascade stage f_r;
- Minimum acceptable true positive ratio per cascade stage d_r;

Initialize:

- Current cascade stage number $t = 0$;
- Current overall cascade classifier's true positive ratio $D_t = 1.0$;
- Current overall cascade classifier's false positive ratio $F_t = 1.0$;
- $S_0 = \{P_0, Q_0\}$ is the initial working set, $Q_0 \subset Q$. The number of positive samples is N_p. The number of negative samples is $N_q = N_p \times R_0, R_0 = 8$;

While $F_t > F_r$

1) $t = t + 1$;$f_t = 1.0$; Normalize the weights $\omega_{t,i}$ for each sample x_i to guarantee that $\omega_t = \{\omega_{t,i}\}$ is a distribution.
2) **While** $f_t > f_r$
   a) For each feature ϕ_m, train a weak classifier on S_0 and find the best feature ϕ_i (the one with the minimum classification error).
   b) Add the feature ϕ_i into the strong classifier H_t, update the weight in Gentle AdaBoost manner.
   c) Evaluate on S_0 with the current strong classifier H_t, adjust the rejection threshold under the constraint that the true positive ratio does not drop below d_r.
   d) Decrease threshold until d_r is reached.
   e) Compute f_t under this threshold.

   END While
3) $F_{t+1} = F_t \times f_t$; $D_{t+1} = D_t \times d_r$; keep in Q_0 the negative samples that the current strong classifier H_t misclassified (current false positive samples), record its size as K_{fq}.
4) Use the detector H_t to bootstrap false positive samples from negative Q randomly and repeat until the negative working set has N_q samples.

END While

Output:

A t-levels cascade where each level has a strong boosted classifier with a set of rejection thresholds for each weak classifier.

Algorithm 1: Initial learning

4.2 Dynamic learning

Once a cascade of peak frame detectors is learned in the initial learning stage (Section 4.1), we are able to enlarge the positive set to increase the discriminative performance of the whole classifier. The AU frames detector will become stronger as new AU positive samples are added for training. We added additional constraint to avoid adding ambiguous AU frames to the dynamic positive set. The algorithm is summarized in Algorithm 2.

Input

- Cascade detector H_0, from the Initial Learning step;
- Dynamic working set $S_D = \{P_D, Q_D\}$;
- All the frames in this action unit are represented as potential positive samples in the set $P = \{P_s, P_v\}$. P_s contains the strong positive samples, P_0 contains peak related samples described above, $P_0 \in P_s$. P_v contains ambiguous positive samples;
- A large negative data set Q contains all other AUs and non-AUs and its size is N_a.

Update positive working set by spreading within P and update negative working set by bootstrap in Q with the dynamic cascade learning process:

Initialize: We set the value of N_p as the size of P_0. The size of the old positive data set is $N_{p\_old} = 0$ and the diffusion stage is $t = 1$.

While $(N_p - N_{p\_old})/N_p > 0.1$

1) **AU Positive Spreading:** $N_{p\_old} = N_p$. Use the current detector on the data set P to potentially add more positive samples, P_{sp} are all the positive samples that are determined by the cascade classifier H_{t-1}.
2) **Hard Constrain in spreading:** k indexes the current AU event and i is the index to the current frame in this AU event. Calculate the similarity values (Eq. 1) between the peak frame in event k and all peak frames with the lowest intensity value 'A', and denote the average similarity value with S_k. Calculate the similarity value between frame i and peak frame in event k, its value is S_{ki}, if $S_{ki} < 0.5 \times S_k$, frame i will be excluded from P_{sp}.
3) After the above step, the remaining positive work set is $P_w = P_{sp}$, $N_p =$ size of P_{sp}. Using the H_{t-1} detector to bootstrap false positive samples from the negative set Q until the negative working set Q_w has $N_q = \beta \times R_0 \times N_a$ samples, N_a is different in AUs.
4) Train the cascade classifier H_t with the dynamic working set $\{P_w, Q_w\}$. As R_t varies, the maximum acceptable false positive ratio per cascade stage f_{m_r} also becomes smaller (Eq. 2).
5) $t = t + 1$; empty P_w and Q_w.

END While

Algorithm 2: Dynamic learning

The function to measure the similarity between the peak and other frames is the Radial Basis function between the

appearance representation of two frames:

$$S_k = \frac{1}{n}\sum_{j=1}^{n} Sim(\boldsymbol{f}_k, \boldsymbol{f}_j), \quad j \in [1:n]$$
$$S_{ik} = Sim(\boldsymbol{f}_i, \boldsymbol{f}_k) = e^{-(Dist(i,k)/max(Dist(:,k)))^2}$$
$$Dist(i,k) = \left[\sum_{j=1}^{m}(f_{kj} - f_{ij})^2\right]^{1/2}, j \in [1:m] \quad (1)$$

n refers to the total number of AU instances with intensity 'A', and m is the length of the AU features.

The dynamic positive working set becomes larger but the negative samples pool is finite, so f_{m_r} needs to be changed dynamically. Moreover, N_q is function of N_a because different AU has different size of the negative samples pool. Some AUs (e.g., AU12) are likely to occur more often than others. Rather than tuning these thresholds one by one, we assume that the false positive ratio f_{m_r} changes exponentially in each stage t, that is:

$$f_{m_r} = f_r \times (1 - e^{-\alpha R_t}) \quad (2)$$

In our experimental set up, we set α as 0.2 and β as 0.04 respectively. We found empirically that those values are suitable for all the AUs to avoid lack of useful negative samples in RU-FACS database. After the spreading stage, the ratio between positive and negative samples becomes balanced, except for some rare AUs (e.g., AU4, AU10) where the unbalance is due to the scarceness of positive frames in the database.

5 Experiments

DCBB iteratively samples training frames and then uses Cascade AdaBoost for classification. To evaluate the efficacy of iteratively sampling training samples, Experiment 1 compared DCBB with the two standard approaches. They are selecting only peaks and alternatively selecting all frames between onsets and offsets. Experiment 1 thus evaluated whether iteratively sampling training images increased AU detection. Experiment 2 evaluated the efficacy of Cascade AdaBoost relative to SVM when iteratively sampling training samples. In our implementation, DCBB uses Cascade AdaBoost, but other classifiers might be used instead. Experiment 2 informs whether better results might be achieved using SVM. Experiment 3 explored the use of three appearance descriptions (Gabor, SIFT and DAISY) in conjunction with DCBB (Cascade Adaboost as classifier).

5.1 Database

The three experiments all used RU-FACS (a.k.a. M3) database [18, 19]. RU-FACS consists of video-recorded interviews of 34 men and women of varying ethnicity. Interviews were approximately two minutes in duration. Video from five subjects could not be processed for technical reasons (e.g., noisy video), which resulted in usable data from 29 participants. Meta-data included manual FACS codes for AU onsets, peaks and offsets. Because some AUs occurred too infrequently, we selected the 13 AUs that occur more than 20 times in the database (i.e. 20 or more peaks). These AUs are: AU1, AU2, AU4, AU5, AU6, AU7, AU10, AU12, AU14, AU15, AU17, AU18, and AU23. Fig. 6 shows the number of frames that each AU occurred and their average duration. Fig. 7 illustrates a representative time series for several AUs from subject S010. Blue asterisks represent onsets, red circles peak frames, and green plus signs the offset frames. In each experiment, we randomly selected 19 subjects for training and the other 10 subjects for testing.

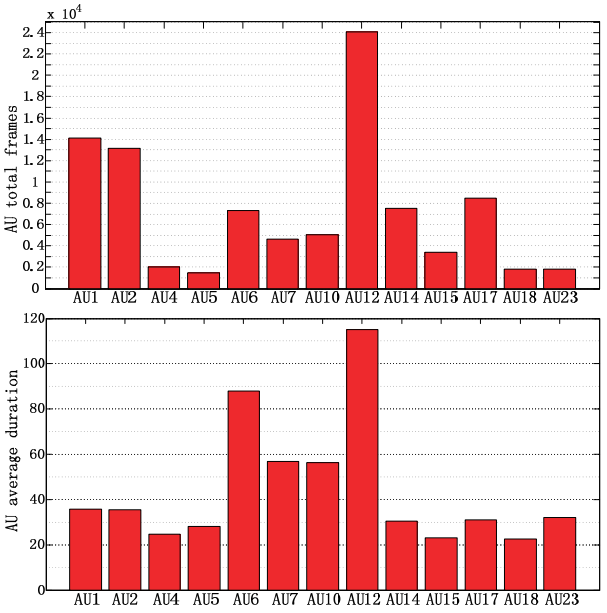


Fig. 6. Top) Frame number from onset to offset for the 13 most frequent AUs in the M3 dataset. Bottom) The average duration of AU in frames.

5.2 Experiment 1: Iteratively sampling training images with DCBB

This experiment illustrates the effect of iteratively selecting positive and negative samples (i.e. frames) while training the Cascade AdaBoost. As a sample selection mechanism on top of the Cascade AdaBoost, DCBB could be applied to other classifiers as well. The experiment investigates whether the iteratively selecting training samples strategy is better than the strategy using only peak AU frames or the strategy using all the AU frames as positive samples. For different positive samples assignments, the negative samples are defined by the method used in Cascade AdaBoost.

We apply the DCBB method, described in Section 4 and use appearance features based on SIFT descriptors (Section 3). For all AUs the SIFT descriptors are built using a square of 48×48 pixels for twenty feature points for the lower face AUs or sixteen feature points for upper face (see Fig. 4). We trained 13 dynamic cascade classifiers, one for each AU, as described in Section 4.2, using a one versus all scheme for each AU.

Top of Fig. 8 shows the manual labeling for AU12 of the subject $S015$. We can see eight instances of AU12 with

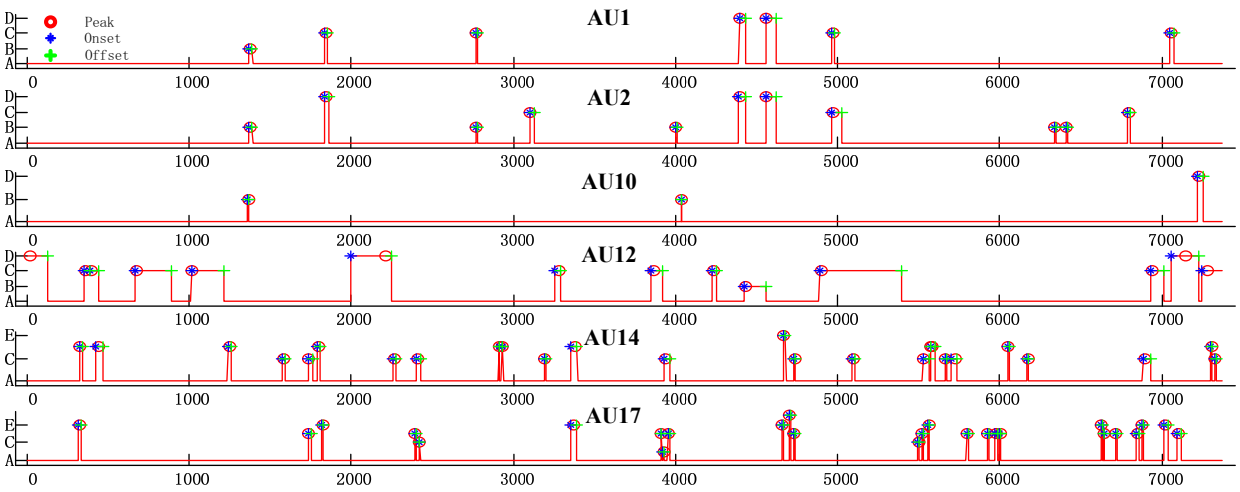


Fig. 7. AUs characteristics in subject S010, duration, intensity, onset, offset, peak.(frames as unit in X axis, Y axis is the intensity of AUs)

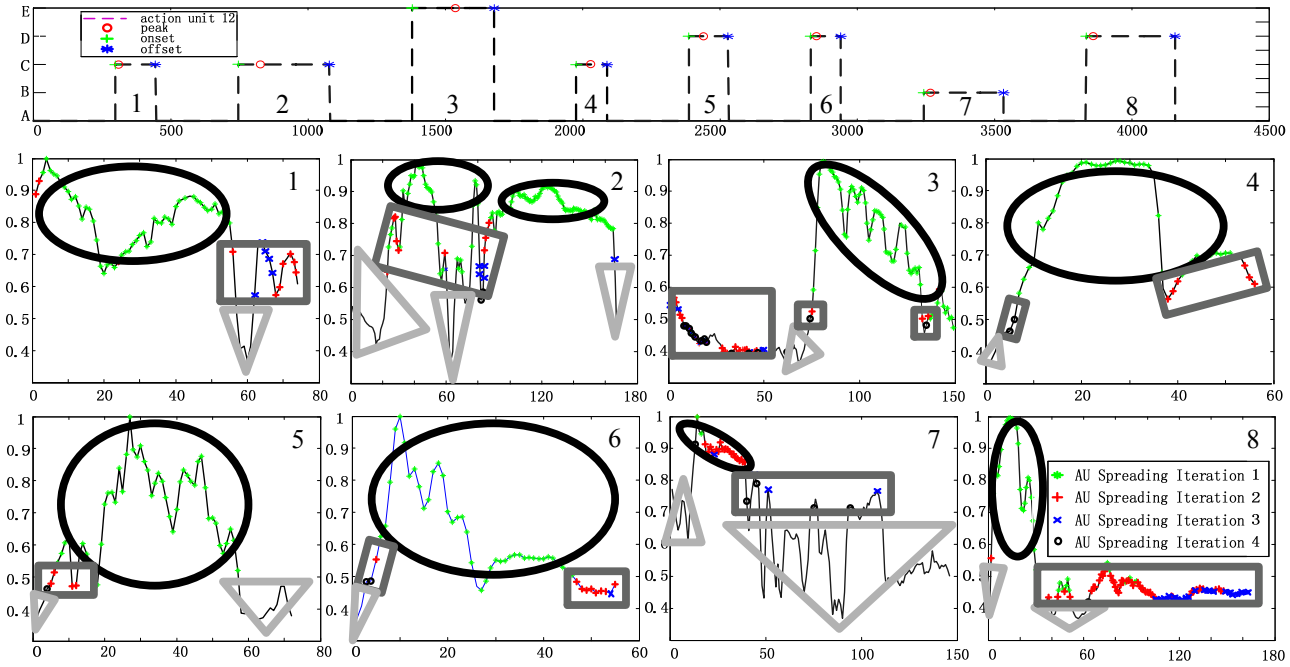


Fig. 8. The spreading of positive samples during each dynamic training step for AU12. See text for the explanation of the graphics.

varying intensities ranging from A (weak) to E (strong). The black curve in bottom figures represent the similarity (eq. 1) between the peak and the neighboring frames. The peak is the maximum of the curve. The positive samples in the first step are represented by green asterisks, in the second iteration by red crosses, in the third iteration by blue crosses, and in the final iteration by black circles. Observe that in the case of high peak intensity, subfigures 3 and 8 (top right number in the similarity plots), the final selected positive samples contain areas of low similarity values. When AU intensity is low, subfigure 7, positive samples are selected if they have a high similarity with the peak, which reduces to the number of false positives. The ellipses and rectangles in the figures contain frames that are selected as positive samples, and correspond to strong and subtle AUs defined above. The triangles correspond to frames between the onset and offset that are not selected as positive samples, and represent ambiguous AUs in Fig. 1.

Table 1 shows the number of frames at each level of intensity and the percentage of each intensity that were selected as positive by DCBB in the training set. The letters ‘A-E’ in left column refers to the level of AU intensities, The ‘Num’ and ‘Pct’ in the top rows refers to the number

TABLE 1
Number of frames at each level of intensity and the percentage of each intensity that were selected as positive by DCBB

	AU1		AU2		AU4		AU5		AU6		AU7	
	Num	Pct	Num	Pct	Num	Pct	Num	Pct	Num	Pct	Num	Pct
A	986	46.3%	845	43.6%	325	37.1%	293	86.4%	119	40.2%	98	47.9%
B	3513	83.3%	3130	79.4%	912	70.4%	371	97.7%	1511	71.3%	705	93.4%
C	3645	90.9%	3241	91.3%	439	83.8%	110	99.7%	1851	87.8%	1388	97.9%
D	2312	90.3%	1753	95.6%	73	79.6%	0	NaN	1128	84.2%	537	98.7%
E	151	99.3%	601	96.5%	0	NaN	23	96.6%	118	91.4%	153	98.5%

	AU10		AU12		AU14		AU15		AU17		AU18		AU23	
	Num	Pct	Num	Pct	Num	Pct	Num	Pct	Num	Pct	Num	Pct	Num	Pct
A	367	61.3%	1561	38.1%	309	49.7%	402	72.6%	445	34.1%	134	87.4%	25	77.0%
B	2131	90.8%	4293	73.3%	2240	93.9%	1053	89.7%	2553	70.1%	665	99.2%	190	87.8%
C	1066	92.8%	5432	82.0%	1040	95.6%	611	97.4%	1644	76.4%	240	98.7%	198	97.3%
D	572	88.6%	2234	83.5%	268	96.8%	68	98.6%	966	85.3%	83	98.6%	166	98.9%
E	0	NaN	1323	92.4%	232	94.3%	0	NaN	226	80.2%	0	NaN	361	99.1%

of AU frames at each level of intensity and the percentage of frames that were selected as positive examples at the last iteration of dynamic learning step, respectively. If there were no AU frames in one level of intensity, the 'Pct' value will be 'NaN'. From this table, we can see that DCBB emphasizes intensity levels 'C', 'D' and 'E'.

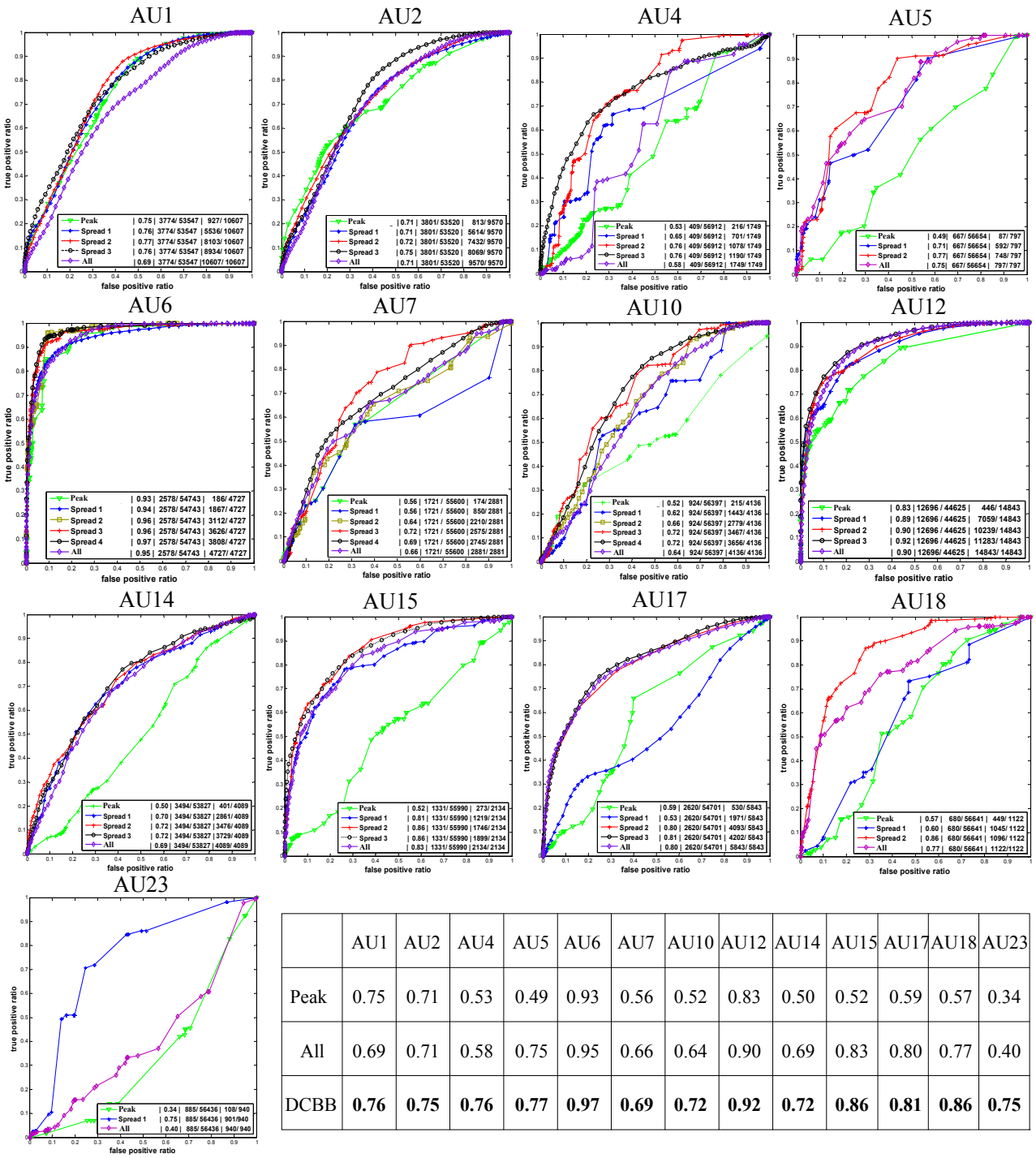
Fig. 9 shows the Receiver-Operator Characteristic (ROC) curves for testing data (subjects not in the training) using DCBB. The ROC curves were obtained by plotting true positives ratios against false positives ratios for different decision threshold values of the classifier. Results are shown for each AU. In each figure, five or six ROC curves are shown: *initial learning* corresponds to training on only the peaks (which is same as Cascade Adaboost without the DCBB strategy); *spread x* corresponds to running DCBB x times; *All* denotes using all frames between onset and offset. The first number between lines | in Fig. 9 denotes the area under the ROC; the second number is the size of positive samples in the testing dataset; and separated by / is the number of negative samples in the testing dataset. The third number denotes the size of positive samples in training working sets and separated by / the total frames of target AU in training data sets. AU5 has the minimum number of training examples and AU12 has the largest number of examples. We can observed that the area under the ROC for frame-by-frame detection improved gradually during each learning stage; performance improved faster for AU4, AU5, AU10, AU14, AU15, AU17, AU18, AU23 than for AU1, AU2, AU6 and AU12 during Dynamic learning. Note that using all the frames between the onset and offset ('All') typically degraded detection performance. The table below Fig. 9 shows the areas under the ROC when only peak is selected for training, when all frames are selected and DCBB is used. DCBB outperformed both alternatives.

Two parameters in the DCBB algorithms were manually tuned and remained the same in all experiments. One parameter specifies the minimum similarity value below which no positive sample will be selected. The threshold is based on the similarity equation (eq. 1). It was set at 0.5 after preliminary testing found results stable within a range of 0.2 to 0.6. The second parameter is the stopping criterion. We consider that the algorithm has converged when the number of new positive samples between iterations is less than 10%. In preliminary experiments, values less than 15% failed to change detection performance as indicated by ROC curves. Using the stopping criterion, the algorithm typically converged within three or four iterations.

5.3 Experiment 2: Comparing Cascade Adaboost and Support Vector Machine (SVM) when used with iteratively sampled training frames

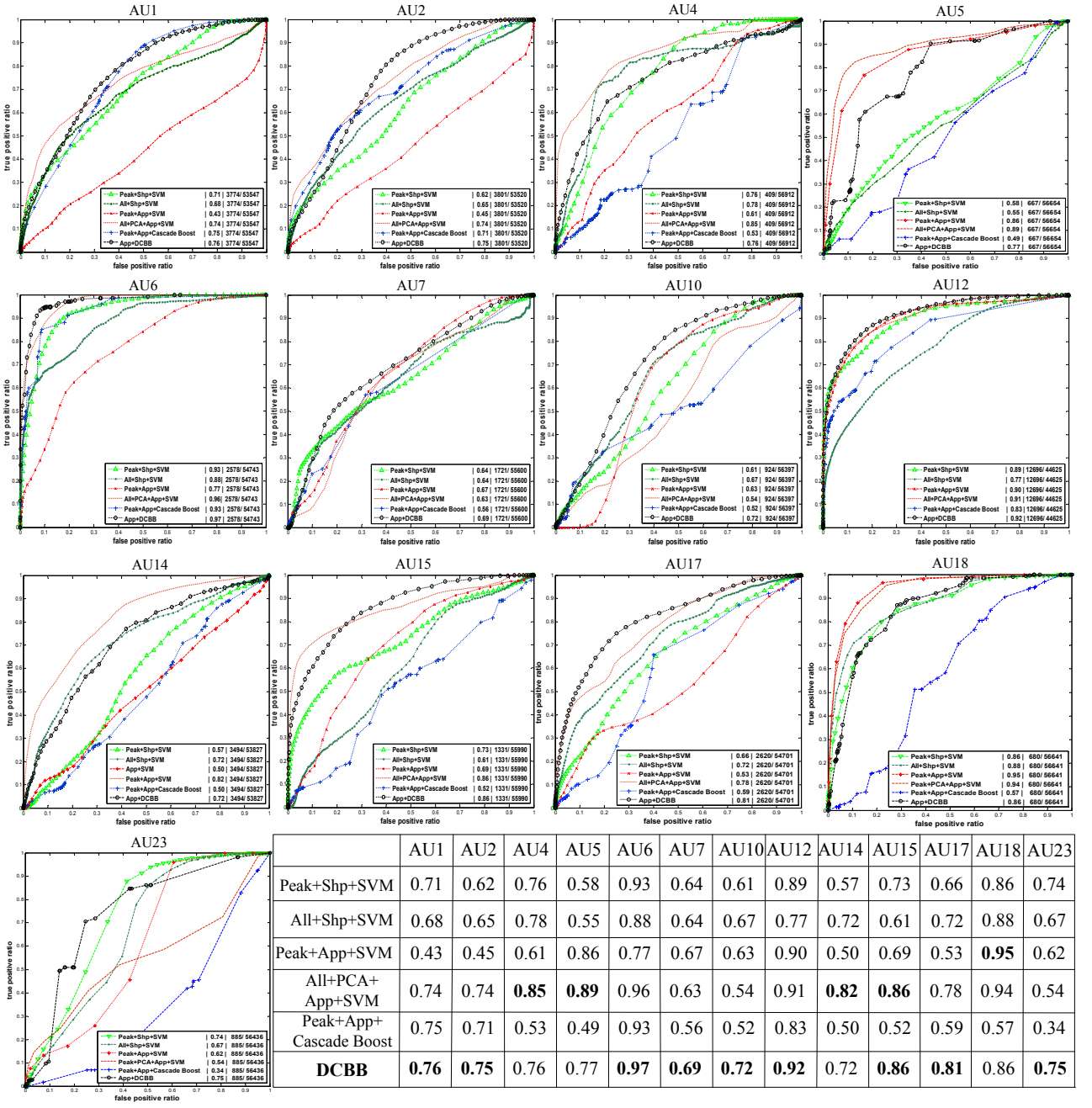
SVM and AdaBoost are two commonly used classifiers for AU detection. In this experiment, we compared AU detection by DCBB (Cascade AdaBoost as classifier) and SVM using shape and appearance features. We found that for both types of features (i.e. shape and appearance), DCBB achieved more accurate AU detection.

For compatibility with the previous experiment, data from the same 19 subjects as above were used for training and the other 10 for testing. Results are reported for the 13 most frequently observed AU. Other AU occurred too infrequently (i.e. fewer than 20 occurrences) to obtain reliable results and thus were omitted. Classifiers for each AU were trained using a one versus-all strategy. The ROC curves for the 13 AUs are shown in Fig. 10. For each AU, six curves are shown, one for each combination of training features and classifiers. 'App+DCBB' refers to DCBB using appearance features; 'Peak+Shp+SVM' refers to SVM using shape features trained on the peak frames (and two adjacent frames) [10]; 'Peak+App+SVM' [10] refers to train a SVM using appearance features trained on the peak frame (and two adjacent frames); 'All+Shp+SVM' refers to SVM using shape features trained on all frames between onset and offset; 'All+PCA+App+SVM' refers to SVM using appearance features (after PCA processing) trained on all frames between onset and offset, here, in order to computationally scale in memory space, we reduced the dimensionality of the appearance features using principal component analysis (PCA) that preserves 98% of the energy. 'Peak+App+Cascade Boost' refers to use



	AU1	AU2	AU4	AU5	AU6	AU7	AU10	AU12	AU14	AU15	AU17	AU18	AU23
Peak	0.75	0.71	0.53	0.49	0.93	0.56	0.52	0.83	0.50	0.52	0.59	0.57	0.34
All	0.69	0.71	0.58	0.75	0.95	0.66	0.64	0.90	0.69	0.83	0.80	0.77	0.40
DCBB	**0.76**	**0.75**	**0.76**	**0.77**	**0.97**	**0.69**	**0.72**	**0.92**	**0.72**	**0.86**	**0.81**	**0.86**	**0.75**

Fig. 9. The ROCs improve with the spreading of positive samples: See text for the explanation of Peak, Spread x and All.



	AU1	AU2	AU4	AU5	AU6	AU7	AU10	AU12	AU14	AU15	AU17	AU18	AU23
Peak+Shp+SVM	0.71	0.62	0.76	0.58	0.93	0.64	0.61	0.89	0.57	0.73	0.66	0.86	0.74
All+Shp+SVM	0.68	0.65	0.78	0.55	0.88	0.64	0.67	0.77	0.72	0.61	0.72	0.88	0.67
Peak+App+SVM	0.43	0.45	0.61	0.86	0.77	0.67	0.63	0.90	0.50	0.69	0.53	**0.95**	0.62
All+PCA+ App+SVM	0.74	0.74	**0.85**	**0.89**	0.96	0.63	0.54	0.91	**0.82**	**0.86**	0.78	0.94	0.54
Peak+App+ Cascade Boost	0.75	0.71	0.53	0.49	0.93	0.56	0.52	0.83	0.50	0.52	0.59	0.57	0.34
DCBB	**0.76**	**0.75**	0.76	0.77	**0.97**	**0.69**	**0.72**	**0.92**	0.72	**0.86**	**0.81**	0.86	**0.75**

Fig. 10. ROC curve for 13 AUs using six different methods: AU peak frames with shape features and SVM(Peak+Shp+SVM); all frames between onset and offset with shape features and SVM (All+Shp+SVM); AU peak frames with appearance features and SVM (Peak+App+SVM); sampling 1 frame in every 4 frames between onset and offset with PCA to reduce appearance dimensionality and SVM (All+PCA+App+SVM); AU peak frames with appearance features and Cascade AdaBoost(Peak+App+Cascade Boost); DCBB with appearance features(DCBB).

the peak frame with appearance features and Cascade AdaBoost [14] classifier (will be equivalent to the first step in DCBB). As can be observed in the figure, DCBB outperformed SVM for all AUs (except AU18) using either shape or appearance when training in the peak (and two adjacent frames). When training the SVM with shape and using all samples, the DCBB performed better for eleven out of thirteen AUs. In the SVM training, the negative samples were selected randomly (but the same negative samples when using either shape or appearance features). The ratio between positive and negative samples was fixed to 30. Compared with the Cascade AdaBoost (first step in DCBB that only uses the peak and two neighbor samples), DCBB improved the performance in all AUs.

Interestingly, the performance for AU4, AU5, AU14, AU18 using the method 'All+PCA+App+SVM' was better than 'DCBB'. 'All+PCA+App+SVM' uses appearance features and all samples between onset and offset. All parameters in SVM were selected using cross-validation. It is interesting to observe that AU4, AU5, AU14, AU15 and AU18 are the AUs that have very few total training samples (only 1749 total frames for AU4, 797 frames for AU5, 4089 frames for AU14, 2134 frames for AU15, 1122 frames for AU18), and when having very little training data the classifier can benefit from using all samples. Moreover, the PCA step can help to remove noise. It is worth pointing out that best results were achieved by the classifiers using appearance (SIFT) instead of shape features, which suggests SIFT may be more robust to residual head pose variation in the normalized images. Using unoptimized MATLAB code, training DCBB typically required one to three hours depending on the AU and number of training samples.

5.4 Experiment 3: Appearance descriptors for DCBB

Our findings and those of others suggest that appearance features are more robust than shape features in unposed video with small to moderate head motion. In the work described above, we used SIFT to represent appearance. In this section, we compare SIFT to two alternative appearance representations, DAISY and Gabor [24, 41].

Similar in spirit to SIFT descriptors, DAISY descriptors are an efficient feature descriptor based on histograms. They have frequently been used to match stereo images [48]. DAISY descriptors use circular grids instead of the regular grids in SIFT; the former have been found to have better localization properties [49] and to outperform many state-of-the-art feature descriptors for sparse point matching [50]. At each pixel, DAISY builds a vector made of values from the convolved orientation maps located on concentric circles centered on the location. The amount of Gaussian smoothing is proportional to the radius of the circles.

In the following experiment, we compare the performance on AU detection for three appearance representations, Gabor, SIFT and DAISY using DCBB with Cascade Adaboost. Each was computed at the same locations (twenty feature points in lower face, see Fig. 12). The same 19 subjects as before were used for training and 10 for testing. Fig. 11 shows the ROC detection curves for the lower AUs using Gabor filters at eight different orientations and five different scales, DAISY [50] and SIFT [44]. For most of AUs, ROC curves for SIFT and DAISY were comparable. With respect to processing speed, DAISY was faster.

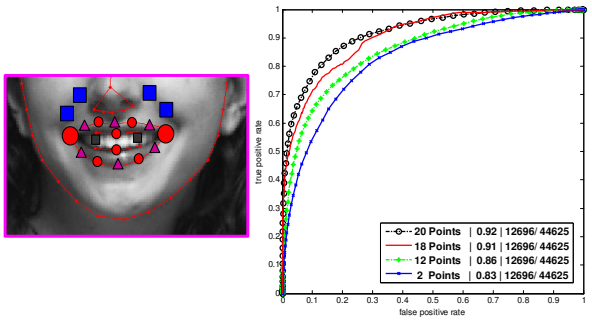
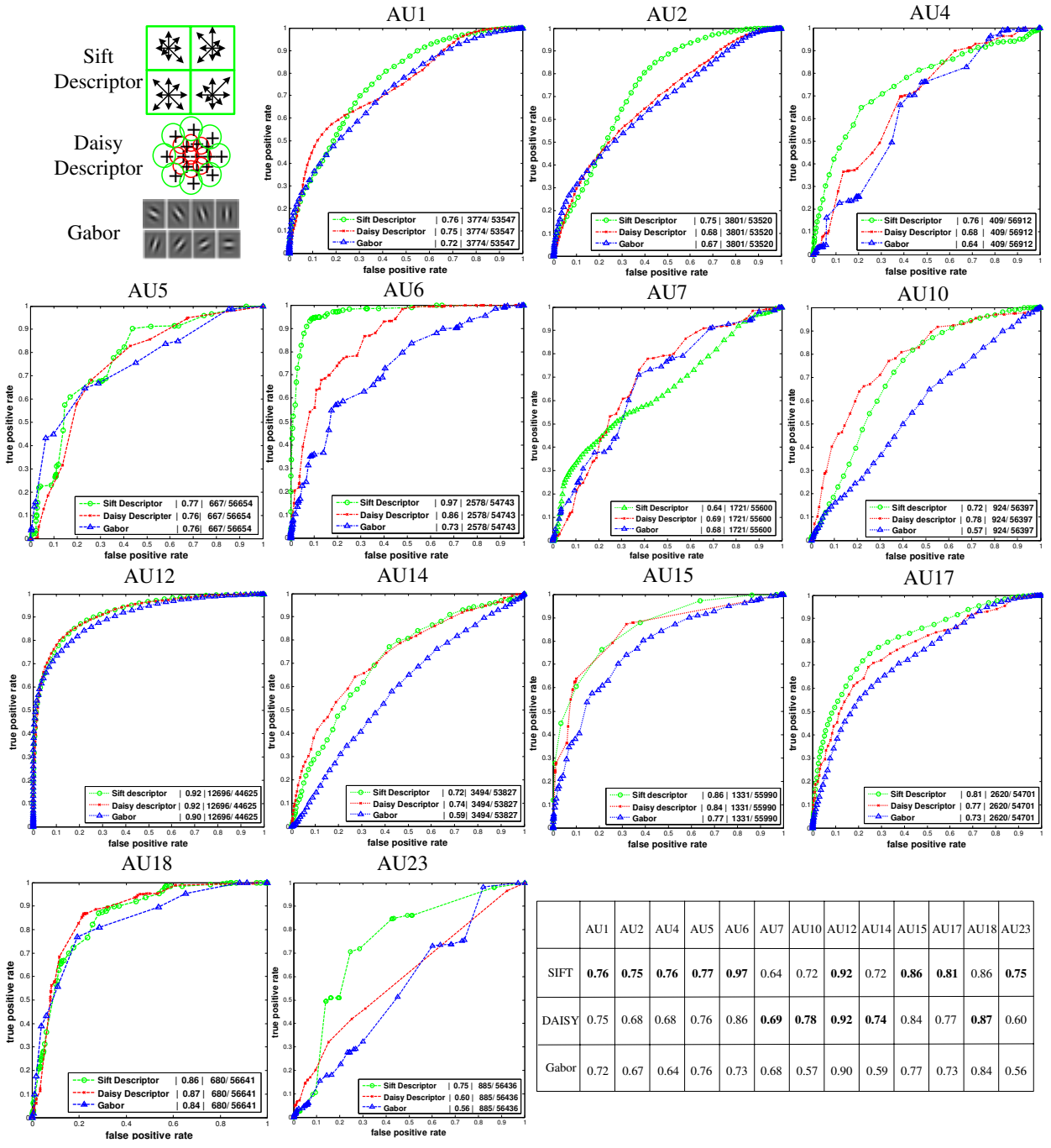


Fig. 12. Using different number of feature points

An important parameter largely conditioning the performance of appearance-based descriptors is the number and location of features selected to build the representation. For computational considerations it is impractical to build appearance representations in all possible regions of interest. To evaluate the influence of number of regions selected, we varied the number of regions, or points, in the mouth area from 2 to 20. As shown in Fig. 12, large red circles represent the location of two feature regions, adding blue squares and red circles increases the number to 12; adding purple triangles and small black squares increases the number to 18 and 20 respectively. Comparing results for 2, 12, 18, and 20 regions, or points, performance was consistent with intuition. While there were some exceptions, in general performance improved monotonically with the number of regions (Fig. 12).

6 Conclusions

An unexplored problem and critical to the success of automatic action unit detection is the selection of the positive and negative training samples. This paper proposes dynamic cascade bidirectional bootstrapping (DCBB) for this purpose. With few exceptions, DCBB achieved better detection performance than the standard approaches of selecting either peak frames or all frames between the onsets and offsets. We also compared three commonly used appearance features and the number of regions of interests or points to which they were applied. We found that use of SIFT and DAISY improved accuracy relative to Gabor under same number of sampling points. For all three, increasing the number of regions or points monotonically improved performance. These findings suggest that DCBB when used with appearance features, especially SIFT or DAISY, can improve AU detection relative to standard selection methods for selecting training samples.



	AU1	AU2	AU4	AU5	AU6	AU7	AU10	AU12	AU14	AU15	AU17	AU18	AU23
SIFT	**0.76**	**0.75**	**0.76**	**0.77**	**0.97**	0.64	0.72	**0.92**	0.72	**0.86**	**0.81**	0.86	**0.75**
DAISY	0.75	0.68	0.68	0.76	0.86	**0.69**	**0.78**	**0.92**	**0.74**	0.84	0.77	**0.87**	0.60
Gabor	0.72	0.67	0.64	0.76	0.73	0.68	0.57	0.90	0.59	0.77	0.73	0.84	0.56

Fig. 11. Comparison of ROC curves for 13 AUs using different appearance representations based on SIFT, DAISY, and Gabor representations

Several issues remain unsolved. The final Cascade Adaboost classifier has automatically selected the features and training samples that improve classification performance in the training data. During the iterative training, we have an intermediate set of classifiers (usually from 3 to 5) that could potentially be used to model the dynamic pattern of AU events by measuring the amount of overlap in the resulting labels. Additionally, our sample selection strategy could be easily applied to other classifiers such as SVM or Gaussian Processes. Moreover, we plan to explore the use of these techniques in other computer vision problems such activity recognition, where the selection of the positive and negative samples might play an important role in the results.

ACKNOWLEDGMENTS

The work was performed when the first author was at Carnegie Mellon University with support from NIH grant 51435. Any opinions, findings and conclusions or recommendations expressed in this material are those of the

authors and do not necessarily reflect the views of the National Institutes of Health. Thanks to Tomas Simon, Feng Zhou, Zengyin Zhang for helpful comments.

REFERENCES

[1] P. Ekman and W. Friesen, "Facial action coding system: A technique for the measurement of facial movement." *Consulting Psychologists Press.*, 1978.

[2] J. F. Cohn, Z. Ambadar, and P. Ekman, *Observer-based measurement of facial expression with the Facial Action Coding System*. New York: Oxford.: The handbook of emotion elicitation and assessment. Oxford University Press Series in Affective Science., 2007.

[3] J. F. Cohn and P. Ekman, "Measuring facial action by manual coding, facial emg, and automatic facial image analysis." *Handbook of nonverbal behavior research methods in the affective sciences.*, 2005.

[4] P. Ekman and E. L. Rosenberg, *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 2005.

[5] M. Pantic, N. Sebe, J. F. Cohn, and T. Huang, "Affective multimodal human-computer interaction." in *ACM International Conference on Multimedia*, 2005, pp. 669–676.

[6] W. Zhao and R. Chellappa, *(Editors). Face Processing: Advanced Modeling and Methods*. Elsevier, 2006.

[7] S. Li and A. Jain, *Handbook of face recognition.* New York: Springer., 2005.

[8] Y. Tian, J. F. Cohn, and T. Kanade, *Facial expression analysis. In S. Z. Li and A. K. Jain (Eds.). Handbook of face recognition.* New York, New York: Springer., 2005.

[9] Y. Tian, T. Kanade, and J. F. Cohn, *Recognizing action units for facial expression analysis*, vol. 23, no. 1, pp. 97–115, 2001.

[10] S. Lucey, A. B. Ashraf, and J. F. Cohn, "Investigating spontaneous facial action recognition through aam representations of the face," in *Face Recognition*, K. D. . M. Grgic, Ed. Vienna: I-TECH Education and Publishing, 2007, pp. 275–286.

[11] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan, "Recognizing facial expression: Machine learning and application to spontaneous behavior," in *CVPR05*, 2005, pp. 568–573.

[12] M. F. Valstar and M. Pantic, "Combined support vector machines and hidden markov models for modeling facial action temporal dynamics," in *ICCV07.*, 2007, pp. 118–127.

[13] Y. Tong, W. H. Liao, and Q. Ji, "Facial action unit recognition by exploiting their dynamic and semantic relationships," *IEEE TPAMI*, pp. 1683–1699, 2007.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR01*, 2001, pp. 511–518.

[15] R. Xiao, H. Zhu, H. Sun, and X. Tang, "Dynamic cascades for face detection," in *ICCV07*, 2007, pp. 1–8.

[16] S. Y. Yan, S. G. Shan, X. L. Chen, W. Gao, and J. Chen, "Matrix-structural learning (msl) of cascaded classifier from enormous training set," in *CVPR07*, 2007, pp. 1–7.

[17] Y. F. Zhu, F. De la Torre, and J. F. Cohn, "Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection," in *Affective Computing and Intelligent Interaction (ACII)*, September 2009.

[18] M. G. Frank, M. S. Bartlett, and J. R. Movellan, "The m3 database of spontaneous emotion expression." 2011.

[19] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Automatic recognition of facial actions in spontaneous expressions." *Journal of Multimedia*, 2006.

[20] M. J. Black and Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," *IJCV97*, vol. 25, no. 1, pp. 23–48, 1997.

[21] F. De la Torre, Y. Yacoob, and L. Davis, "A probabilisitc framework for rigid and non-rigid appearance based tracking and recognition," in *Proceedings of the Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, 2000, pp. 491–498.

[22] Y. Chang, C. Hu, R. Feris, and M. Turk, "Manifold based analysis of facial expression," in *CVPRW04*, 2004, pp. 81–89.

[23] C. Lee and A. Elgammal, "Facial expression analysis using nonlinear decomposable generative models." in *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, 2005, pp. 17–31.

[24] M. Bartlett, G. Littlewort, I. Fasel, J. Chenu, and J. Movellan., "Fully automatic facial action recognition in spontaneous behavior." in *Proceedings of the Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FG'06)*, 2006, pp. 223–228.

[25] M. Pantic and L. Rothkrantz, "Facial action recognition for facial expression analysis from static face images." *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 1449–1461, 2004.

[26] M. Pantic and I. Patras, "Dynamics of Facial Expression: Recognition of Facial Actions and their Temporal Segments from Face Profile Image Sequences," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 36, pp. 433–449, 2006.

[27] P. Yang, Q. Liu, X. Cui, and D. Metaxas, "Facial expression recognition using encoded dynamic features." in *CVPR08*, 2008.

[28] Y. Sun and L. J. Yin, "Facial expression recognition based on 3d dynamic range model sequences," in *ECCV08*, 2008, pp. 58–71.

[29] B. Braathen, M. Bartlett, G. Littlewort, and J. Movellan, "First steps towards automatic recognition of spontaneous facial action units," in *Proceedings of the ACM Conference on Perceptual User Interfaces*, 2001.

[30] P. Lucey, J. F. Cohn, S. Lucey, S. Sridharan, and K. M. Prkachin, "Automatically detecting action units from faces of pain: Comparing shape and appearance features," *CVPRW09*, pp. 12–18, 2009.

[31] J. F. Cohn and T. Kanade, "Automated facial image analysis for measurement of emotion expression," in *The handbook of emotion elicitation and assessment. Oxford University Press Series in Affective Science*, J. A. C. . J. B. Allen, Ed., pp. 222–238.

[32] J. F. Cohn and M. A. Sayette, "Spontaneous facial expression in a small group can be automatically measured: An initial demonstration." *Behavior Research Methods*, vol. 42, no. 4, pp. 1079–1086, 2010.

[33] 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 *CVPRW05*, 2005, pp. 76–83.

[34] M. Valstar and M. Pantic, "Fully automatic facial action unit detection and temporal analysis," in *CVPR06*, 2006, pp. 149–157.

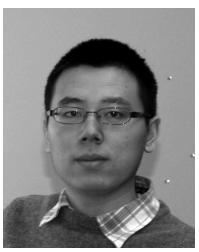
[35] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F. De la Torre, and J. F. Cohn, "Aam derived face representations for robust facial action recognition," in *Proceedings of the Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FG'06)*, 2006, pp. 155–160.

[36] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE TPAMI*, pp. 39–58, 2009.

[37] T. F. Cootes, G. Edwards, and C. Taylor, "Active appearance models," in *ECCV98*, 1998, pp. 484–498.

[38] F. De la Torre and M. Nguyen, "Parameterized kernel

principal component analysis: Theory and applications to supervised and unsupervised image alignment," in *CVPR08*, 2008.
[39] I. Matthews and S. Baker, "Active appearance models revisited," *IJCV04*, pp. 135–164, 2004.
[40] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces." in *SIGGRAPH99*, 1999.
[41] Y. L. Tian, T. Kanade, and J. F. Cohn, "Evaluation of gabor-wavelet-based facial action unit recognition in image sequences of increasing complexity," in *Proceedings of the Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FG'02)*. Springer, 2002, pp. 229–234.
[42] A. B. Ashraf, S. Lucey, T. Chen, K. Prkachin, P. Solomon, Z. Ambadar, and J. F. Cohn, "The painful face: Pain expression recognition using active appearance models," in *Proceedings of the ACM International Conference on Multimodal Interfaces (ICMI'07)*, 2007, pp. 9–14.
[43] P. Lucey, J. F. Cohn, S. Lucey, S. Sridharan, and K. M. Prkachin, "Automatically detecting pain using facial actions." in *International Conference on Affective Computing and Intelligent Interaction (ACII2009)*, 2009.
[44] D. Lowe, "Object recognition from local scale-invariant features," in *ICCV99*, 1999, pp. 1150–1157.
[45] K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE TPAMI*, pp. 39–51, 1998.
[46] L. Breiman, *Classification and regression trees*. Chapman & Hall/CRC, 1998.
[47] R. Schapire and Y. Freund, "Experiments with a new boosting algorithm," in *Machine learning: proceedings of the Thirteenth International Conference (ICML'96)*. Morgan Kaufmann Pub, 1996, p. 148.
[48] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *CVPR08*, 2008.
[49] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE TPAMI*, vol. 27, no. 10, pp. 1615–1630, 2005.
[50] E. Tola, V. Lepetit, and P. Fua, "Daisy: An efficient dense descriptor applied to wide baseline stereo," *IEEE TPAMI*, vol. 99, no. 1, pp. 3451–3467, 2009.



Yunfeng Zhu received the B.Sc. degree in Information Engineering, the M.Sc. degree in Pattern Recognition and Intelligent System from Xi'an Jiaotong University in 2003 and 2006 respectively. He is currently a Ph.D. candidate at Image and Graphics Institute, in the Department of Electronic Engineering at Tsinghua University, Beijing, China. In 2008 and 2009, he became visiting student in the Robotics Institute at Carnegie Mellon University (CMU), Pittsburgh, PA. His research interests include computer vision, machine learning, facial expression recognition and 3D facial model reconstruction.

Fernando De la Torre received the B.Sc. degree in telecommunications and the M.Sc. and Ph.D. degrees in electronic engineering from Enginyeria La Salle in Universitat Ramon Llull, Barcelona, Spain, in 1994, 1996, and 2002, respectively. In 1997 and 2000, he became an Assistant and an Associate Professor, respectively, in the Department of Communications and Signal Theory at the Enginyeria La Salle. Since 2005, he has been a Research Faculty at the Robotics Institute, Carnegie Mellon University (CMU), Pittsburgh, PA. His research interests include machine learning, signal processing, and computer vision, with a focus on understanding human behavior from multimodal sensors. He is directing the Human Sensing Lab (http://humansensing.cs.cmu.edu) and the Component Analysis Lab at CMU (http://ca.cs.cmu.edu). He has co-organized the first workshop on component analysis methods for modeling, classification, and clustering problems in computer vision in conjunction with CVPR-07, and the workshop on human sensing from video in conjunction with CVPR-06. He has also given several tutorials at international conferences on the use and extensions of component analysis methods.

Jeffrey F. Cohn earned the PhD degree in clinical psychology from the University of Massachusetts in Amherst. He is a professor of psychology and psychiatry at the University of Pittsburgh and an Adjunct Faculty member at the Robotics Institute, Carnegie Mellon University. For the past 20 years, he has conducted investigations in the theory and science of emotion, depression, and nonverbal communication. He has co-led interdisciplinary and interinstitutional efforts to develop advanced methods of automated analysis of facial expression and prosody and applied these tools to research in human emotion and emotion disorders, communication, biomedicine, biometrics, and human-computer interaction. He has published more than 120 papers on these topics. His research has been supported by grants from the US National Institutes of Mental Health, the US National Institute of Child Health and Human Development, the US National Science Foundation, the US Naval Research Laboratory, and the US Defense Advanced Research Projects Agency. He is a member of the IEEE and the IEEE Computer Society.

Yu-Jin Zhang received the Ph.D. degree in Applied Science from the State University of Lige, Lige, Belgium, in 1989. From 1989 to 1993, he was post-doc fellow and research fellow with the Department of Applied Physics and Department of Electrical Engineering at the Delft University of Technology, Delft, the Netherlands. In 1993, he joined the Department of Electronic Engineering at Tsinghua University, Beijing, China, where he is a professor of Image Engineering since 1997. In 2003, he spent his sabbatical year as visiting professor in the School of Electrical and Electronic Engineering at Nanyang Technological University (NTU), Singapore. His research interests are mainly in the area of Image Engineering that includes image processing, image analysis and image understanding, as well as their applications. His current interests are on object segmentation from images and video, segmentation evaluation and comparison, moving object detection and tracking, face recognition, facial expression detection/classification, content-based image and video retrieval, information fusion for high-level image understanding, etc. He is vice president of China Society of Image and Graphics and director of academic committee of the Society. He is also a Fellow of SPIE.