MIT. Media Laboratory Affective Computing Technical Report #571

Appears in IEEE International Workshop on Analysis and Modeling of Faces and Gestures , Oct 2003

Fully Automatic Upper Facial Action Recognition

Ashish Kapoor   Yuan Qi   Rosalind W. Picard

MIT Media Laboratory
Cambridge, MA 02139

Abstract

This paper provides a new fully automatic framework to analyze facial action units, the fundamental building blocks of facial expression enumerated in Paul Ekman's Facial Action Coding System (FACS). The action units examined in this paper include upper facial muscle movements such as inner eyebrow raise, eye widening, and so forth, which combine to form facial expressions. Although prior methods have obtained high recognition rates for recognizing facial action units, these methods either use manually preprocessed image sequences or require human specification of facial features; thus, they have exploited substantial human intervention. This paper presents a fully automatic method, requiring no such human specification. The system first robustly detects the pupils using an infrared sensitive camera equipped with infrared LEDs. For each frame, the pupil positions are used to localize and normalize eye and eyebrow regions, which are analyzed using PCA to recover parameters that relate to the shape of the facial features. These parameters are used as input to classifiers based on Support Vector Machines to recognize upper facial action units and all their possible combinations. On a completely natural dataset with lots of head movements, pose changes and occlusions, the new framework achieved a recognition accuracy of 69.3% for each individual AU and an accuracy of 62.5% for all possible AU combinations. This framework achieves a higher recognition accuracy on the Cohn-Kanade AU-coded facial expression database, which has been previously used to evaluate other facial action recognition system.

1 Introduction

A very large percentage of our communication is nonverbal and among these nonverbal cues a large fraction is in the form of facial actions. A system that could analyze the facial actions in real time without any human intervention will have applications in a number of different fields: for example, computer vision, affective computing, computer graphics and psychology. Such a system will be an important component in a machine that is socially and emotionally intelligent and is expected to interact naturally with people.

While a lot of research has been directed towards systems that recognize faces corresponding to prototypic expressions like joy, anger and surprise, few approaches exist that try to recognize facial actions such as eye-squint and frown. Table 1 compares some of the previous facial expression analysis techniques. The state of the art systems have severe limitations as they either require human intervention or do not recognize more than prototypic expressions. This paper describes a fully automatic framework that requires no manual intervention to analyze facial activity. The work is focused on recognizing upper action units(AUs) which are a subset of all the AUs enumerated in Paul Ekman's Facial Action Coding System (FACS) [8] and correspond to the regions of eyes and eyebrows (Table 2). The Facial Action Coding System (FACS) developed by Ekman and Friesen [8] is a method of measuring facial activity in terms of facial muscle movements. FACS consists of over 45 distinct AUs corresponding to a distinct muscle or muscle group and are essentially facial phonemes, which can be assembled to form facial expressions. Finally, most researchers have reported results on clean datasets, which are videos and images of the frontal face of the subjects deliberately making facial actions in front of a camera. We evaluate our framework on a test dataset which is completely natural and therefore has lots of pose changes, head movements, occlusions and very subtle facial activity. To our best knowledge this is the only work that is evaluated on a completely natural database, therefore demonstrating how computer vision and machine learning can be integrated to build real-world applications.

Table 1: Comparison of various face analysis systems

	Fully automatic	**Recognize more than prototype expressions**
Black & Yacoob [2] 1995	No	No
Esaa et al [9] 1997	Yes	No
Tian et al [16] 2000	No	Yes

Table 2: The upper facial action units recognized in this paper

AU number	**Facial action**
1	Inner brow raiser
2	Outer brow raiser
4	Brow lowerer
5	Upper eye lid raiser
7	Lid tightener

2 Previous Work

Researchers in the past have used a number of classification techniques to recognize action units and their combinations. Tian et al [16] have developed a system to recognize sixteen action units and any combination of those. The shape of facial features like eyes, eyebrow, mouth and cheeks are described by multistate templates. The parameters of these multistate templates are used by a Neural Network based classifier to recognize the action units. This system requires that the templates be initialized manually in the first frame of the sequence, which prevents it from being fully automatic. In an earlier work, Lien et al [13] describe a system that recognizes various action units based on dense flow, feature point tracking and edge extraction.

Donato et al [6] compared several techniques, which included optical flow, principal component analysis, independent component analysis, local feature analysis and Gabor wavelet representation, to recognize eight single action units and four action unit combinations using image sequences that were manually aligned and free of head motions. They showed 95.5% recognition accuracy using Independent Component Analysis and Gabor wavelet representations. They have used a nearest neighbor classifier and template matching for the purpose of recognition. Each facial action combination that they try to recognize is treated as a separate AU. As there are a large number of AU combinations, modeling each AU combination separately is not appropriate. Bartlett et al [1] achieve 90.9% accuracy in recognizing 6 single action units by combining holistic facial analysis and optical flow with local feature analysis. Both of the above mentioned approaches [1, 6] report their results on manually pre-processed image sequences of individuals deliberately making facial actions in front of a camera.

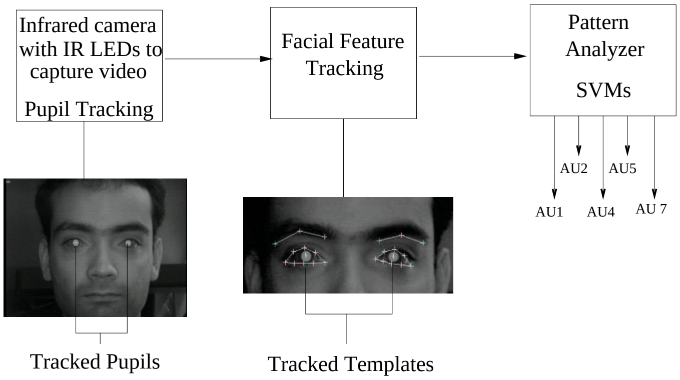


Figure 1: The overall system

Cowie et al [5] describe a system to recognize facial expressions by identifying *Facial Animation Parameter Units (FAPUs)* defined in MPEG-4 standard by feature tracking of *Facial Definition Parameter(FDP)* points, also defined in MPEG-4 framework. The system is not fully automatic and requires human assistance to accurately detect FDP points.

A lot of research has been directed at the problem of recognizing 5-7 classes of prototypic emotional expressions on groups of people from their facial expressions [2, 9, 17, 18]. Although prototypic expressions, like happy, surprise and fear, are natural, they occur infrequently in everyday life. A person might communicate more with subtle facial actions like frequent frowns or smiles. Further there are emotions like confusion, boredom and frustration for which any prototypic expression might not exist. Thus, a system that aims to be socially and emotionally intelligent needs to do more than just recognize prototypic expressions.

3 Overall Framework

Figure 1 gives you an overview of the system. The red-eye effect [10] is a a physiological property of the eye and the first part concerns using it to robustly track the pupils. Once the pupil positions are known, those are then used to normalize the images and extract parameters that describe facial features and can be used to recognize the facial actions. Finally, the upper facial action units are recognized using Support Vector Machines. A separate Fisher kernel is trained for each facial action unit. The extracted parameters are used as input features to the support vector machines to detect occurrence of facial actions. Since we use a separate classifier for each action unit, they can detect action unit combinations.

3.1 Pupil Detection

The pupil detection system detects the pupils using the red-eye effect. The system's robustness to occlusions and head motions makes it ideal to be used for automatic facial action analysis. As the pupil positions can be recovered very efficiently and robustly, it eliminates the need of manual labeling or pre-processing of the images, a required step that plagues a number of previous approaches.

Although the red-eye effect has been known for quite sometime, it is in recent years that it has grabbed a lot of attention for vision applications. Morimoto et al [14] have described a system to detect and track pupils using the red-eye effect. Haro et al [10] have extended this system to detect and track the pupils using a Kalman filter and probabilistic PCA. We use an infrared camera equipped with infrared LEDs, which is used to highlight and track pupils and is an in-house built version of the IBM Blue Eyes camera. The whole unit is placed under the monitor pointing towards the users face. The system has an infrared sensitive camera coupled with two concentric rings of infrared LEDs. One set of LEDs is on the optical axis and produces the red-eye effect. The other set of LEDs, which are off axis, keeps the scene at about the same illumination. The two sets of LEDs are synchronized with the camera and are switched on and off to generate two interlaced images for a single frame. The image where the on-axis LEDs are on has white pupils whereas the image where the off-axis LEDs are on has black pupils. These two images are subtracted to get a difference image, which is used to track the pupils. Figure 2 shows a sample image, the de-interlaced images and the difference image obtained using the system.

The pupils are detected and tracked using the difference image, which is noisy due to the interlacing and motion artifacts. Also, objects like glasses and earrings can show up as bright spots in the difference image due to their specularity. To remove this noise we first threshold the difference image using an adaptive thresholding algorithm [10]. First, the algorithm computes the histogram and then thresholds the image keeping only 0.1 % of the brightest pixels. All the non-zero pixels in the resulting image are set to 255 (maxval). The thresholded image is used to detect and to track the pupils.

3.2 Feature Extraction

For the purpose of facial action analysis, we need to track the facial features robustly and efficiently. Also, rather then just tracking the positions of facial features, we need to recover the parameters that drive the shape of the feature. The variability in appearance of facial features changes due to pose, lighting, facial expressions etc making the task difficult and complex. Even harder is the task of tracking the facial features robustly in real time, without any manual alignment or calibration. Many previous approaches have focused just on tracking the location of the facial features or require some manual initialization/intervention. In this section, we describe how we can robustly recover the shape of facial features in detail using templates in real time without requiring any manual intervention. More details on the pupil and facial feature tracking can be found in [12].

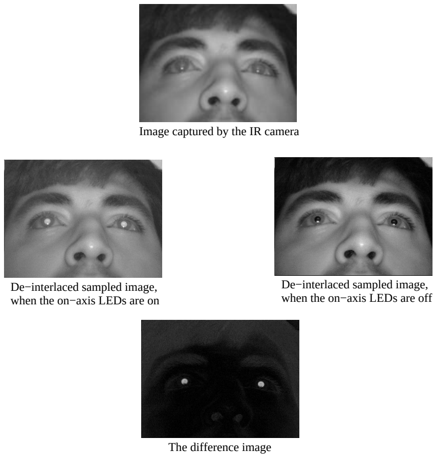


Image captured by the IR camera

De−interlaced sampled image, when the on−axis LEDs are on

De−interlaced sampled image, when the on−axis LEDs are off

The difference image

Figure 2: Pupil tracking using the infrared camera

Our system exploits the fact that it can estimate the location of pupils very robustly in the image. Once the pupils are located, *regions of interest* corresponding to eyes and eyebrows are cropped out and analyzed to recover the shape description. For the purpose of the facial action analysis, the fiducial points of the templates describing eyes and eyebrows are considered as shape parameters. Our goal is then to recover these fiducial points in a new image. Assume, that we have a training set of image vectors $\{\mathbf{i}_1, \mathbf{i}_2, .., \mathbf{i}_n\}$, where each image vector $\mathbf{i}_k$ is pre-annotated with a corresponding vector of shape parameters $\mathbf{s}_k$. For the purpose of facial action analysis, the images $\mathbf{i}$ are cropped images of eyes and eyebrows and the vector of shape parameters $\mathbf{s}$ is a stack of x,y coordinates of fiducial points.

To recover the shape parameters in a test image, say $\mathbf{i}_{test}$, a very naive approach will be to find an image, $\mathbf{i}_{match}$, from the training set of pre-annotated images that most closely resembles $\mathbf{i}_{test}$. The shape parameters of $\mathbf{i}_{test}$ then can be approximated by the shape parameters $\mathbf{s}_{match}$, which corresponds to $\mathbf{i}_{match}$. This approach cannot generalize well, as there can be only a finite number of example images in the training database. A more general approach will be to represent the test image as a linear combination of example images. The same linear combination can be applied to the corresponding shape parameters of the example images to recover the shape in the new image. Principal component analysis (PCA) can be used to figure out the representation of the test image in terms of the linear combination of example images. Given n example images $\mathbf{i}_k$, let $\mathbf{s}_k$ $(k = 1..n)$ be vectors corresponding to the marked control points on each image. If $\bar{\mathbf{i}}$ is the mean image, then the covariance matrix of the training images can be expressed as:

$$\mathbf{\Lambda} = \mathbf{P} \cdot \mathbf{P}^T \text{ where } \mathbf{P} = [\mathbf{i}_1 - \bar{\mathbf{i}}, \mathbf{i}_2 - \bar{\mathbf{i}}, ..., \mathbf{i}_n - \bar{\mathbf{i}}]$$

The eigenvectors of $\mathbf{\Lambda}$ can be computed by first computing the eigenvectors for $\mathbf{P}^T \cdot \mathbf{P}$. If $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..\mathbf{v}_n]$ where $\mathbf{v}_k$ represents the eigenvectors of $\mathbf{P}^T \cdot \mathbf{P}$, then the eigenvectors $\mathbf{u}_k$ of $\mathbf{\Lambda}$ can be computed as:

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n] = \mathbf{P} \cdot \mathbf{V}$$

As the eigenvectors are expressed as a linear combination of example images, we can express the shape parameters corresponding to the eigen images using the same linear combination. Let $\bar{\mathbf{s}}$ be the mean of the vectors corresponding to the control points in example images and let $\mathbf{Q} = [\mathbf{s}_1 - \bar{\mathbf{s}}, ..., \mathbf{s}_n - \bar{\mathbf{s}}]$, be the matrix of unbiased shape parameters. Then, the shape parameters $\tilde{\mathbf{s}}_k$ $(k = 1..n)$ corresponding to an eigenvector $\mathbf{u}_k$ can be computed as:

$$[\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, ..., \tilde{\mathbf{s}}_n] = \mathbf{Q} \cdot \mathbf{V}$$

To recover the shape parameters in the test image, we first express the new image as a linear combination of the eigenvectors by projecting it onto the top few eigenvectors.

$$\mathbf{i}_{new} = \sum_k a_k \mathbf{u}_k + \bar{\mathbf{i}} \tag{1}$$

where $a_k = (\mathbf{i}_{test} - \bar{\mathbf{i}})^T \cdot \mathbf{u}_k$ and $\mathbf{u}_k$ is the k^{th} eigenvector.The same linear combination is applied to the shape parameters of corresponding eigenvectors to recover the new shape.

$$\mathbf{s}_{new} = \sum_k a_k \tilde{\mathbf{s}}_k + \bar{\mathbf{s}} \tag{2}$$

In brief, the training set is first processed offline to compute the required eigenvectors. During the real-time tracking the cropped images of the eyes and eyebrows are projected on the corresponding top few eigenvectors. Experiments showed that the first 40 eigenvectors suffice for the task and in our implementation we use those. These projections are used to recover the control points as explained above. This approach worked particularly well on the subjects who had their images in the training database. This strategy is a simplification of the approach used by Covell et al[4] and performs well for the purpose of the facial feature tracking, particularly on the subjects who had images in the training set. Note that there is no initialization step, which was very critical in many template matching approaches. Further, the non-iterative nature of the approach makes it ideal to be used in a real-time system.

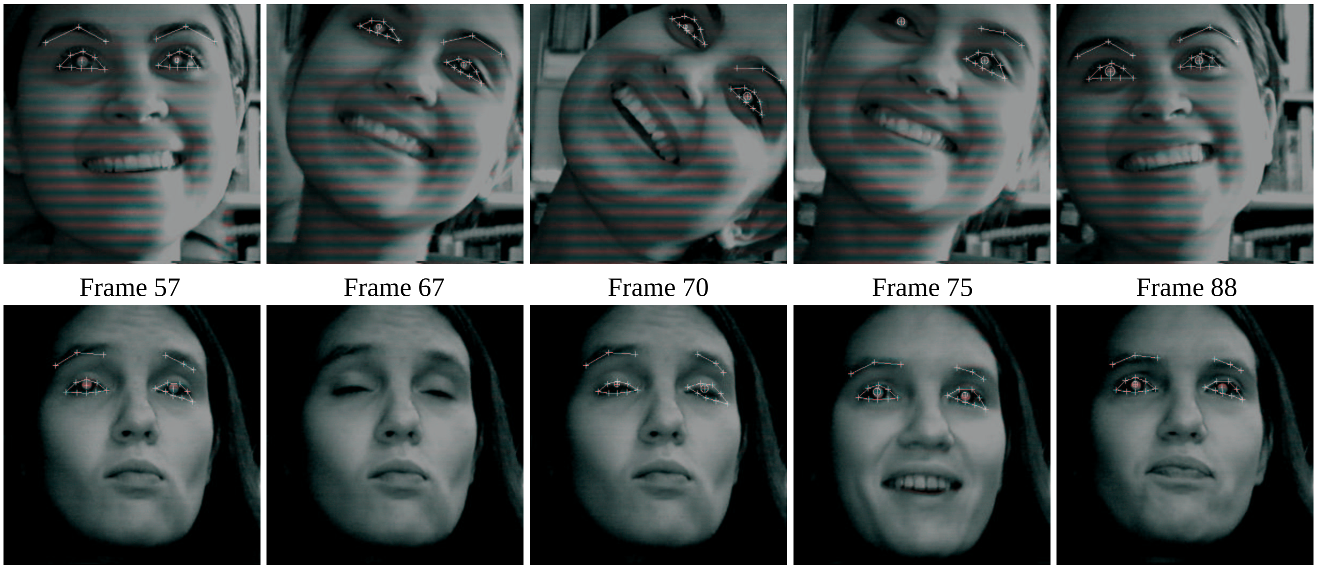
This approach is very efficient, runs in real time at 30 fps and is able to track upper facial features robustly in presence of large head motions and occlusions. One limitation of our implementation is that it is not invariant to large zooming in or out as fixed size images of the facial features are cropped. Further, our training set did not have samples with scale changes. Also in a few cases with some new subjects, the system did not work well, as the training images were not able to span the whole range of variations in appearance of the individuals. A training set which captures the variations in appearance should be able to overcome these problems.

Figure 3 shows tracking results of some sequences. The first subject appearing in Figure 3 was in the training database. The system is able to track the features very well. Note that in the first sequence of Figure 3 the left eyebrow is not tracked in frames 67, 70 and 75 as it is not present in the image. Similarly all the templates are lost in the frame 29 in the second sequence of Figure 3 when the pupils are absent, as the subject blinks. The templates are recovered as soon as the features reappear. The second sequence in Figure 3 shows the tracking results for a subject not in the training set. Again, note that the second frame in the first sequence does not show any eyes or eyebrows, due to the fact that the subject blinked and hence no pupils were detected. The tracking recovers in the very next frame when the pupils are visible again.

Despite various advantages, this strategy has some shortcomings. It assumes a linear relationship between the image and the shape parameters, which might not be the case. Also, it uses principal component analysis to recover the shape; hence, it assumes that the top eigenvectors capture the shape variations, which is erroneous. There may be variations due to lighting which would contribute highly to the principal components.

3.3 Action Unit Classification

Once the parameters are extracted the next step is to identify the facial actions they correspond to. There are lots of classifiers that could be used for the purpose of AU recognition. Support Vector Machines (SVM) have been shown to perform well on a number of classification tasks. Classifiers based on SVM perform binary classification by first projecting the data points onto a linearly separable feature space and then, using a hyperplane that is maximally separated from the nearest positive and negative data points. Math-



Frame 57 Frame 67 Frame 70 Frame 75 Frame 88

Frame 59 Frame 60 Frame 61 Frame 68 Frame 87

Figure 3: Upper Facial Feature Tracking.

ematically, given a set of N training data (x_i, y_i), where $x_i \in \mathbb{R}^d$ with corresponding label $y_i \in \{1, -1\}$, the support vector machine classifies a data point x using,

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i k(x, x_i) - b$$

Here $k(x, x_i)$ is a positive definite kernel function and specifies an inner product between x and x_i in the linearly separable feature space. The x_i's corresponding to non-zero α_i's are support vectors. The α_i's and the bias b can be obtained by solving an optimization problem. For the purpose of classifying facial actions, x is the vector of relevant shape parameters and the sign of $f(x)$ determines whether an AU has been recognized or not.

There are over 40 different facial action units enumerated in FACS [8] and more than 7,000 different AU combinations have been observed. A system that aims to analyze faces should not only recognize a single AU but the combinations of AUs as well. The AU combinations can be additive or non-additive. The appearance of AUs does not change when additive combination of AUs occur, whereas in non-additive combinations, the appearance of individual AUs does change. In our system a separate SVM for each AU is trained using examples. We have used Cawley's SVM toolbox [3] to train the SVMs to classify the facial feature parameters that correspond to an occurrence of a particular AU from the ones that don't. During the recognition phase, the facial feature parameters in each frame are first normalized to account for variability due to change in pose, zoom and personal variations. These normalized parameters are then subtracted from the normalized parameters corresponding to a neutral frame. This difference is used as input features to the SVMs to figure out which AUs were present. Also rather than using all the shape parameters, we use only those that are most indicative of the action unit that we are trying to recognize. We use parameters that describe eyebrows to recognize AU 1, 2 and 4 and the eye parameters for AU 5 and 7. The next section describes evaluation of the system and the results.

4 Experimental Evaluation Results

The system was evaluated on two datasets. The first dataset is completely natural with lots of head motions, occlusions and pose changes. The results are indicative of how a system like this would work in real-world applications. The second database we use is the Cohn-Kanade database [11], which has been previously used to test facial action recognition systems. The results on this database enable comparison between the framework here and other published methods that have been trained and tested on the Cohn-Kanade database.

The natural facial action database has 8 children in a real learning situation. These children were asked to play a game called the *fripple place* [7]. The game has a number of puzzles that requires mathematical reasoning. Each kid worked on these puzzles for about 20 minutes. Videos of their faces were recorded by two cameras. A vision camera was placed on top of the monitor and an IBM Blue Eyes camera was placed under the monitor. A FACS trained expert coded the videos of the face for various action units and 80 frames were selected from these FACS coded videos of the kids. These frames were selected manually to ensure that there were an equal number of samples of the different facial action units from all the kids. The Cohn-Kanade database is a comprehensive database collected and coded by a team of researchers and consists of adults performing a series of facial expressions in front of a camera. The training and testing database (CMU database) has video sequences of 25 individuals, each video sequence starting with a neutral frame. The details of both the datasets are shown in Table 3.

Table 3: Details of instances of AUs in the datasets

Action Unit	# of instances in our database	# of instances in CMU database
1	35	37
2	33	27
4	16	58
5	24	21
7	13	27
Neutral	19	27
Total	**140**	**197**

Table 4: Results for individual AUs in our database

Action Unit	# of Samples	Correct Recognition	Misses	% Correct Recognition
AU 1	35	26	9	74.3%
AU 2	33	26	7	78.8%
AU 4	16	9	7	56.2%
AU 5	24	16	8	66.7%
AU 7	13	6	7	46.1%
Neutral	19	14	5	73.3%
Total	**140**	**97**	**43**	**69.29%**

Table 5: Results for individual AUs in CMU database

Action Unit	# of Samples	Correct Recognition	Misses	% Correct Recognition
AU 1	37	27	10	73.0%
AU 2	27	25	2	92.6%
AU 4	58	47	11	81.0%
AU 5	21	14	7	66.7%
AU 7	27	27	0	100.0%
Neutral	27	20	7	100.0%
Total	**197**	**160**	**37**	**81.22%**

The system is evaluated for recognition accuracy using leave-one-subject-out cross validation. The classifiers were trained using the data from all but one subject and reserving the one subject for testing. This was repeated for all subjects in the database. The system could recognize each individual AU with an accuracy of 69.29%, whereas an accuracy of 62.5% was obtained for all AU combinations. Table 4 shows how well each individual AU was recognized and Table 6 shows how well each AU combination was recognized. Although the results are significantly better than random, they are not as high as we would like to attain; moreover, they are lower than what has been previously reported in the literature. However, it is important to keep in mind that these results are calculated on a dataset arising from natural human behavior; this set is very different from the datasets used to evaluate earlier systems. The videos have a lot of occlusion and head movements, which makes the problem much harder than on datasets where the expressions are largely staged, and the images pre-processed and manually normalized.

The system was also evaluated on the CMU database. For evaluation purposes, the pupils in the CMU database were hand marked as the database was not shot using a Blue Eyes camera and the pupil positions could not be extracted automatically. Further, rather than using the extracted facial features, we used the PCA coefficients computed separately for eye and eyebrow regions. Note that the PCA coefficients are linearly related to the facial feature parameters. On the CMU database the system could recognize each individual AU with an accuracy of 81.22%. Table 5 shows the complete results. The results are comparable to results previously reported on the same database by Tian et al [15]. A lot of earlier work in face analysis reported very high recognition results and at first glance the results reported here on the natural database might seem insignificant. But, we have to keep in mind that most of the earlier work has focused on frontal video of the face shot in ideal conditions. The systems were trained and tested at the apex of emotional expression and required human intervention to identify the neutral and apex frames prior to processing. Considering that an accuracy of 75% among the human FACS coders is required for certification as an expert, the new fully-automatic system performance is comparable to that of human experts. In real-world applications the face

Table 6: Results for AU combinations in our database

Actual AUs	# of Samples	Fully Recognized	Partially Recognized	% Full Correct
1+2	12	9	1	75%
1+2+5	19	11	3	57.9%
1+2+7	2	1	1	50%
1+4	2	0	2	0%
4	10	5	0	50%
5	5	5	0	100%
7	7	3	0	42.9%
4+7	4	2	1	50%
Neutral	19	14	0	73.7%
Total	**80**	**50**	**8**	**62.5%**

analysis system should be fully automatic and should not require any human intervention, which is challenging due to the presence of head movements, pose variations and occlusions in a natural scenario. This system is evaluated in these challenging conditions; hence, the results are state of the art for vision applied to natural human behavior.

5 Conclusion and Future Work

This paper demonstrates a fully automatic framework that can recognize upper facial action units. This framework can be used in scenarios where the machine needs a perceptual ability to recognize, model and analyze the facial activity in real time without any manual intervention. The system first tracks the pupil positions robustly using the red-eye effect; these positions are then used to localize eyes and eyebrows. The shape parameters corresponding to these facial features are recovered using Principal Component Analysis (PCA). Once the parameters describing the facial features are recovered, they are used to recognize the facial actions. Support vector machines (SVMs) are used to recognize facial actions and a recognition accuracy of 69.29% for each individual AU is reported. The system can correctly identify all possible AU combinations with an accuracy of 62.5% in a real and fully natural dataset. The dataset used for evaluation is completely natural and the paper demonstrates how computer vision and machine learning can be integrated to build real-world applications.

The framework suggested in this paper has some limitations. The system depends upon the robust pupil tracking, which currently breaks when the subjects are wearing glasses. The pattern recognition to find pupils can be further refined to track the pupils even when there are subjects with glasses. Since the system uses infrared LEDs, it can be confused by the presence of strong direct sunlight as in an automobile (although we had no problems with indirect daylight from an office window); consequently, this would need modification for some environments. It is also possible to refine the shape parameter extraction by taking into account zoom and variations due to pose changes. The system can also be extended to track lower facial features, like the lips and nose, and to recognize lower facial action units as well. The face is a very important channel that emits signals related to the internal state and a lot of effort is being devoted to unravel this relationship. Besides being used as a man-machine interface, this framework would hopefully be useful to a lot of these research efforts as well .

Acknowledgments

We thank Jeffrey F. Cohn and Zara Ambadar for providing the Cohn-Kanade FACS database. This research was supported by NSF ROLE grant 0087768.

References

[1] M. A. Bartlett, J. C. Hager, P. Ekman, and T. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36(2):253–263, March 1999.

[2] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric model of image motion. In *Proceedings of the International Conference on Computer Vision*, pages 374–381, Cambridge, MA, 1995. IEEE Computer Society.

[3] G. C. Cawley. MATLAB support vector machine toolbox (v0.50β) [http://theoval.sys.uea.ac.uk/˜gcc/svm/toolbox]. University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000.

[4] Michele Covell. Eigen-points: control-point location using principal component analyses. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, October 1996.

[5] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor. Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine*, 18(1):33–80, January 2001.

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

[7] Edmark. Fripple place. http://www.riverdeep.net/edconnect/softwareactivities /critical_thinking/fripple_place.jhtml.

[8] P. Ekman and W. V. Friesen. *The Facial Action Coding System: A Technique for Measurement of Facial Movement*. Consulting Psychologists Press, San Francisco, CA, 1978.

[9] I. Essa and A. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *Pattern Analysis and Machine Intelligence*, 7:757–763, July 1997.

[10] A. Haro, I. Essa, and M. Flickner. Detecting and tracking eyes by using their physiological properties. In *Proceedings*

of Conference on Computer Vision and Pattern Recognition, June 2000.

[11] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, 2000.

[12] Ashish Kapoor and Rosalind W. Picard. Real-time, fully automatic upper facial feature tracking. In *Proceedings of Conference on Automatic Face and Gesture Recognition*, May 2002.

[13] J. Lien, T. Kanade, J. Cohn, and C. C. Li. Detection, tracking and classification of action units in facial expression. *Journal of Robotics and Autonomous Systems*, 31:131–146, 2000.

[14] C. Morimoto, D. Koons, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. Technical report, IBM Almaden Research Center, 1998.

[15] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing upper face action units for facial expression analysis. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, June 2000.

[16] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence*, 23(2), February 2001.

[17] Y. Yacoob and L. Davis. Computing spatio-temporal representation of human faces. In *CVPR*, pages 70–75, Seattle, WA, June 1994.

[18] Z. Zhang. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron. *International Journal of Pattern Recognition and Artificial Intelligence*, 13(6):893–911, 1999.