

# Automatic Detection of Facial Feature Points in Image Sequences

Rajesh A Patil  
MNIT Jaipur  
Email:rapatil\_rtg@yahoo.co.in

Vineet Sahula  
MNIT Jaipur  
Email:sahula@ieee.org

A. S. Mandal  
CEERI Pilani  
Email:atanu@ceeri.ernet.in

**Abstract**—Detection and location of the face as well as extraction of facial features from images is an important stage for numerous facial image interpretation tasks. Detection of facial feature points, such as corners of eyes, lip corners, nostrils from the images are crucial. In this paper a method for automatic facial feature point detection in image sequences, is introduced. The method uses image normalization, and thresholding techniques to detect 14 facial feature points. Algorithm proposed by Wolf Kienzle is used for face recognition. The detected face region is then divided into 5 relevant regions of interest, each of which is examined separately, further to detect the location of the facial feature points. In each region image is normalized with respect to brightness. Suitable threshold is set for each region, using which image is converted into binary image. Then in each region extreme ends of binary image will locate the facial feature points. In the eye region horizontal and vertical histograms are analyzed to detect eyeballs. This method when tested on Cohn-Kanade database results in recognition rate of 86%. Moreover, when tested on Informatics and Mathematical Modeling (IMM) face database which consists of tilted faces around y axis, we achieved average recognition rate of 83%.

**Index Terms**—Feature detection, Threshold, Normalization.

## I. INTRODUCTION

Computer vision is the science or technology of machine to extract information from an image, to solve some task, or to understand the scene. It is a difficult task - a task that seems relatively trivial to humans is actually complex for computers to perform. Computer vision research concerns, not only understanding the process of vision, but also designing effective vision systems for various real world applications. In such applications the face identification, facial expression recognition, face tracking, lip reading detection of facial feature points is the first step. Active appearance algorithm and active shape models have been widely used for face alignment and tracking, for which locating facial feature points is very important. Facial feature points normally include eyes corners, eyebrows corners, nostrils, iris, eyeballs, lip corners. In many applications this step is usually carried out by manually labeling the required facial feature points.

The methods available for facial feature point detection are texture-based and shape-based methods. Local texture around a given feature point, is considered in texture-based methods, for example the pixel values in a small region around a eye corner. While in shape based methods all facial feature points are treated as a shape, which is learned from a set of labeled faces, and try to find the proper shape for any unknown face. Gray-value, eye-configuration and neural-

network-based eye-feature detection [1], log Gabor wavelet based facial point detection [2], and two-stage facial point detection using a hierarchy of Gabor wavelet networks [3] are the typical texture based methods. Typical shape-based methods include active appearance model based facial feature detectors [4], [5].

Reinders et al. [1] propose a feature detector that use Neural network. To detect an eye from images, they scan an input window of  $i*j$  pixels across the image, where each gray value in the input window serves as an input for the neural network. This network with  $i*j$  inputs, a hidden layer and one output, is then trained to give a high response when the input window is centered on the eye. After scanning the entire image, the position with the highest response then reveals the center position of the eye in the image. The method is tested on only eyes and requires rigorous training.

Holden et al. [2] propose automatic facial feature point detection using log-Gabor wavelet. The technique specifically identifies the facial points required for lip and head trackers. The technique detects seven facial feature points, namely the outer eye corners, one nostril, and the outer corners and top and bottom mid points of the lips. Each facial point is represented with a jet of 24 dimensional log Gabor wavelet responses, which uses log-Gabor filter covering six directions and four scales. They solve the feature correspondence problem by employing geometric cross-ratios, which are invariant to 2D rotations and scaling.

Feris et al. [3] present a technique for facial feature localization using a two level hierarchical wavelet network. They construct a training database of face images and their two level Gabor wavelet networks representations. The first level Gabor wavelet networks, representing the entire face, is used to find a face in the database that is similar to the target, and to determine an affine transformation to describe any difference in the orientation of the faces. The second level Gabor wavelet networks, representing each feature, are initialized in positions according to the affine transformation from the first level Gabor wavelet network. They are then allowed to move slightly to minimize their difference from the new face. This facilitates adjustments to account for slight differences in the geometry of the database face and the target. The final position of the child wavelet networks is the estimate of the feature positions. They detect eight facial feature points eyes corners, nostrils and lip corners.

Hu et al. [4] proposed a method, called Active Wavelet

Networks, in which a Gabor wavelet network representation is used to model the texture variation in the training set. The Gabor wavelet network approach represents a face image through a linear combination of 2D Gabor functions whose parameters position, scale, orientation, and weights are optimally determined to preserve the maximum image information for a chosen number of wavelets.

Yan et al. [5] propose appearance model, called Direct appearance model (DAM). Its extended view based models are applied for multi view face alignment. Similar to Active appearance model (AAM), DAM makes use of both shape and texture constraints, however it does not combine them as in AAM. Texture information is directly used to predict the shape and estimate the position and appearance.

A number of other approaches combining texture and shape-based methods have also been proposed. Wiskott et al. [6] used Gabor jet detectors and modeled the distribution of facial features with a graph structure. They use the phase of the complex Gabor wavelet coefficients to achieve a more accurate location of the nodes and to disambiguate patterns which would be similar in their coefficient magnitudes. They employ object adapted graphs, so that nodes refer to specific facial landmarks, called fiducial points. The correct correspondences between two faces can then be found across large viewpoint changes. They use a data structure, called the bunch graph, which serves as a generalized representation of faces by combining jets of a small set of individual faces. This allows the system to find the fiducial points in one matching process, which eliminates the need for matching each model graph individually.

Cristinacce and Cootes [7] used Haar feature based AdaBoost classifier combined with the statistical shape models. They use local AdaBoost templates to localize each feature that are swift to compute. A shape model is used to constrain the configuration of a set of candidate feature points. They introduce a search algorithm, that allows the decision on facial features to be grouped and tested. This method avoids local minima, by searching many possible feature point configurations. The method allows for feature detector failures by predicting the positions of missing features using the shape model. Only eyes and lip corners are detected.

Chen et al. [8] proposed a method that applies a boosting algorithm to determine facial feature point candidates for each pixel in an input image and then uses a shape model as a filter to select the most possible position of feature points, like eyes, nostrils and lip corners.

D. Vukadinovic et al. [9] propose a method using Gabor wavelets and AdaBoost classifier. They detect 20 facial feature points in images of expressionless face. The method consists of four steps: face detection, region of interest detection (ROI), feature extraction, and feature classification. For face detection they use fast and robust face detector based on a cascade scheme consisting of a set of Harr feature based Gentleboost classifier. The detected face region is then divided into 20 ROIs, each consisting of one facial feature using a combination of heuristic techniques based on the analysis of vertical and horizontal image histograms. Each ROI is filtered

by a set of 48 Gabor filters, 6 scales and 8 orientations. For a certain facial feature point  $13 \times 13$  pixels window is slid pixel by pixel across 49 representations [gray scale plus 48 Gabor filter representations] of the relevant ROI. For each position of the sliding window, Gentleboost classifier gives a response depicting the similarity between the 49 representations of the sliding window compared to the learned feature point model. After scanning the entire ROI, the position with the highest response detects the feature point. As they have considered  $13 \times 13$  pixel window,  $169 \times 49 = 8281$  features are used to represent one point. For each feature point they have used 9 positive and 16 negative samples for training, which requires a  $25 \times 8281$  size matrix to represent training data for each ROI for each training image. The method is applicable only for expressionless faces. It works only for the frontal view faces. They detect 20 feature points with 93% accuracy. Author has tested it on images from Cohn Kanade database and they can not guarantee that the accuracy will remain same for other databases.

In general, although some of these detectors seem to perform quite well when localizing a small number of facial feature points such as the corners of the eyes and the mouth, none of them detects all facial feature points and, more importantly, none performs the detection with high accuracy [9].

We propose a method for detecting 14 facial feature points in any image from image sequence with tilted faces under varying illumination conditions. The method is comprised of 5 steps: Face detection, Region of interest detection, image normalization, thresholding, and corner detection. To locate face region in an input image we use face detection algorithm proposed by Wolf Kienzle [10]. Viola Jones [11] algorithm is also available for face detection, but we got 99% accuracy using Wolf Kienzle algorithm for the database which we are using. While using Viola Jones algorithm we got 95% accuracy. Time required to detect face using Wolf Kienzle is 0.7 seconds, while Viola Jones algorithm needs 0.9 seconds. So we decided to use Wolf Kienzle algorithm. After detecting face region, it is divided horizontally into three parts, such that upper region contains eyes, middle contains nostrils, and lower contains mouth. Then the upper region is divided vertically into two parts such that each one will contain one eyes. Each eye region is again divided horizontally into two parts so that eyes and eyebrows are separated. In eye regions horizontal and vertical histogram analysis is performed and peaks are found to detect eyeballs. In each region we perform image normalization and thresholding so that image is converted into binary image. Then corners are detected in order to get desired features. Our method detects feature points from expressionless as well as faces with expressions such as smile, surprise, sad etc. It is tested for tilted faces also.

The rest of the paper is organized as follows. In Section II we describe different steps of our method. Section III describes the experimental results obtained by our method. In Section IV we conclude.

## II. FACIAL FEATURE POINTS DETECTION SYSTEM

### A. Face detection

To detect facial feature points automatically, it is first necessary to locate face in the image. For face detection we make use of a face detection scheme proposed by Wolf Kienzle [10]. They have used the technique of Viola-Jones [12] algorithm, but used SVM classifier instead of AdaBoost classifier.

### B. Detecting Regions of Interest

The face detection algorithm detects the face as shown in Figure 1(a). Face is enclosed by a rectangle. This algorithm

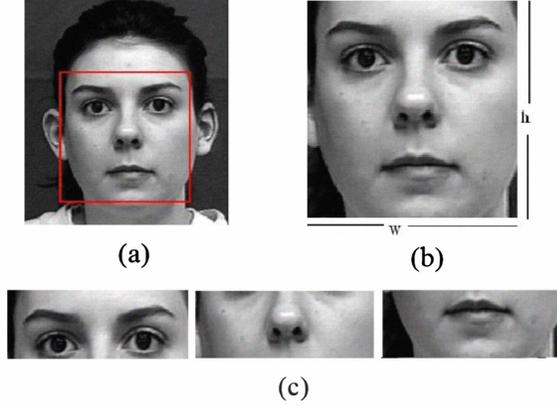


Figure 1. (a) Face detection, (b) Cropped face image, (c) Face divided into three parts

gives the coordinates of upper left corner of rectangle. It also gives the width ( $w$ ) and height ( $h$ ) of the rectangle as shown in Figure 1(b). First we crop the image according to the rectangle. Now we divide this image into three equal parts horizontally as shown in Figure 1(c), so that upper part contains eyes, middle part contains nostrils, and lower part contains mouth. Again the upper part is divided into two parts vertically, so that each part contains one eye, as shown in Figure 2(a).



Figure 2. (a) Left and right eyes separated, (b) Eyebrow and eye separated

Again each eye region is divided horizontally into two parts so that eyebrows and eyes are separated as shown in Figure 2(b). The position of the eyeball from the segmented eye region can be detected by sequentially applying analysis of the vertical histogram which shows the intensity differences between the successive rows pixel wise. The peak of this histogram gives the  $y$  coordinate of the eyeball. Similarly peak of horizontal histogram which shows the intensity differences between the successive columns pixel wise gives  $x$  coordinate of the eyeball. With this approach we have detected eyeballs as shown in Figure 3.



Figure 3. Eyeballs detected

### C. Feature Detection

1) *Eye corner detection:* We consider each of the segmented eyes region separately. For the left eye, image is cropped from left side, while for right eye image is cropped from right side. Then for eye region we apply image normalization technique, which is a linear process. For example if the intensity range of the image is 50 to 180 and the desired range is 0 to 255, the process entails subtracting 50 from each of pixel intensity, making the range 0 to 130. Then each pixel intensity is multiplied by  $255/130$ , making the range 0 to 255. If  $N_1 \times N_2$  is the size of the image, then we compute average using (1).

$$avg = \frac{1}{N_1 N_2} \sum_{x=1}^{N_1} \sum_{y=1}^{N_2} R(x, y) \quad (1)$$

Here  $R(x, y)$  is pixel value at coordinate  $(x, y)$ . Then, we select a suitable threshold value. After trial and error method we have selected threshold value as  $\frac{avg}{3}$ . Using this threshold gray scale image is converted into binary image as in (2).

$$if R(x, y) > \frac{avg}{3}, R(x, y) = 1, else R(x, y) = 0 \quad (2)$$

Following thresholding we get binary image. Resulting binary image is scanned vertically to get first and last black point, which actually compound to corners of eyes as shown in Figure 4(a).

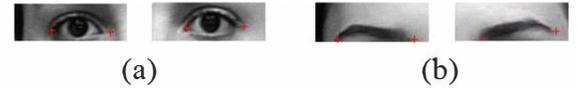


Figure 4. (a) Eyes corner detected, (b) Eyebrows corner detected

2) *Eyebrows corner detection:* Now we consider another ROI that is eyebrows region. Left side of the left eye, and right side of right eye is cropped. Here also we apply image normalization, thresholding and vertical scanning technique to get eyebrows corners. Eyebrows are detected as shown in Figure 4(b).

3) *Nostrils Detection:* Now we consider next ROI that is nostril region. Again the same techniques are applied in this region to detect nostrils, as shown in Figure 5(a).

4) *Lip corner detection:* Now we consider the mouth region. In this region also we crop image from left as well as from right side. Then we apply image normalization, thresholding, and vertical scanning techniques to detect lip corners as shown in Figure 5(b).

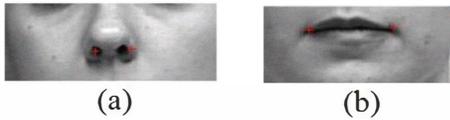


Figure 5. (a) Nostril detection, (b) Lip corner detection

### III. EXPERIMENTAL RESULTS

The proposed algorithm has been tested on Cohn-Kanade [13] database as well as on Informatics and Mathematical Modeling (IMM) [14] database. We have detected 14 facial feature points with average accuracy of 86% on Cohn-Kanade database and 83% on IMM face database. Some of the results are shown in Figure 6.

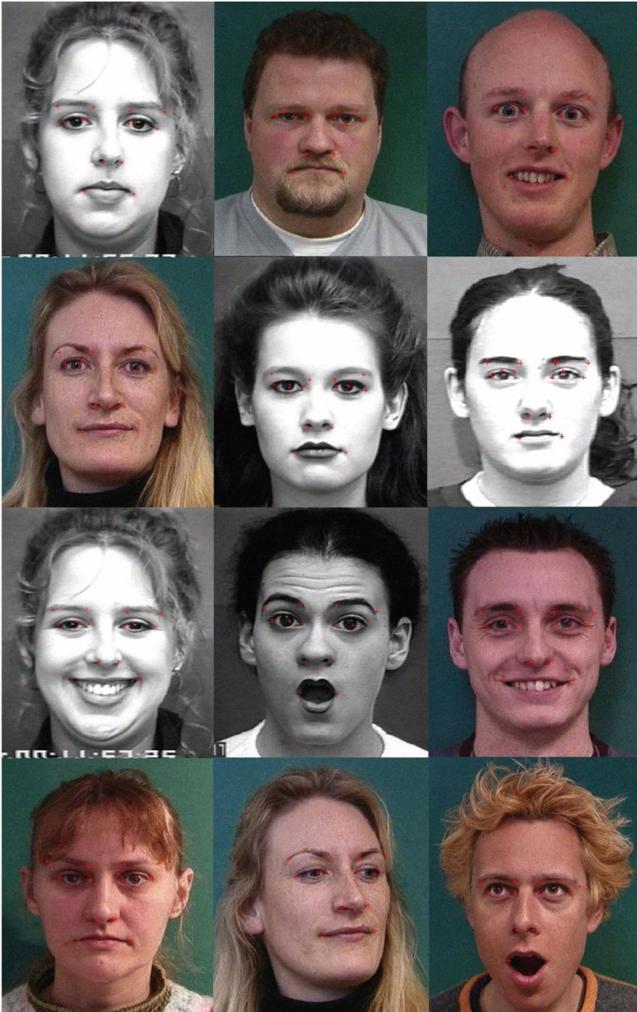


Figure 6. Accurate detection of all facial points

Exact points are not located, but approximately we have located 14 facial feature points. In many applications such as Candide wire frame model fitting on face image, approximate location of facial feature points is required. In comparison with the previously used techniques such as Gabor filters [9], we used very simple technique. So computational complexity is very less. In [9] they used Gabor filters and AdaBoost algorithm to detect 20 facial feature points. They divides

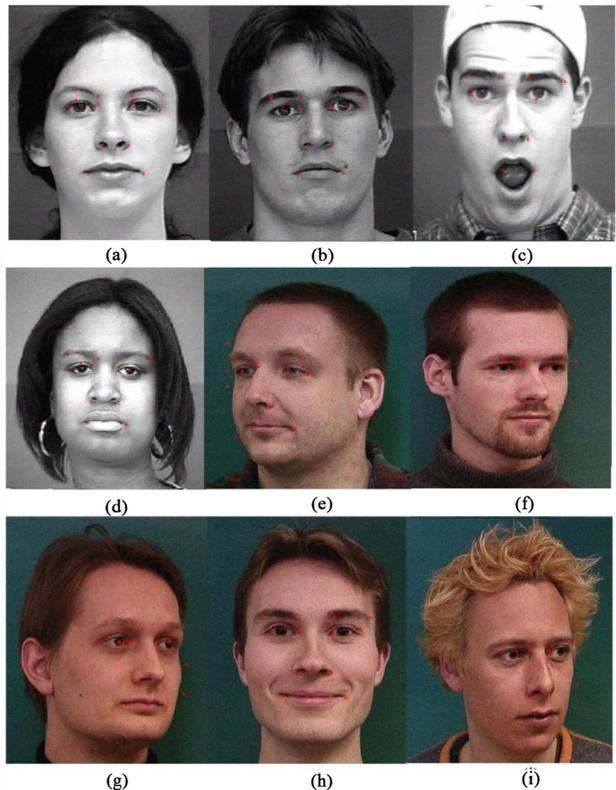


Figure 7. Inaccurate detection of facial points: (a) Points E1, E2, C1, D2, (b) Points C1, A1, (c) Points A2, B1, B2, E2, G1, (d) Points E1, F2, (e) Points C1, A1, E1, E2, D2, G1, G2, (f) Points C1, E1, E2, D2, F2, (g) Points C1, C2, A1, D1, D2, B1, B2, F1, F2, (h) Points E2, F2, (i) Points C1, C2, A1, A2, D1, D2, B2, E2, F2

face image into 20 ROI. Each ROI is convolved with set of 48 Gabor filters, and then Gabor coefficients are collected as features in a huge matrix form, which increases computational complexity as well as computational time. For training  $25 \times 8281$  size matrix represents one ROI. In the testing phase  $13 \times 13$  pixel window is slid pixel by pixel across 49 representations of the ROI. Then Gentleboost classifier classifies the pixel as a feature or non feature. As they are using AdaBoost classifier, they need training, which is a time consuming process. Their method is applicable only for expressionless and frontal facial images. Even after huge computations the technique is applicable only for Cohn-Kanade database. They got 93% average accuracy. In our case accuracy is 85%, but computational complexity is less. No training data is required. It can be applied for tilted faces with expressions. Cohn Kanade database consists of mainly frontal view faces. IMM database consists of frontal as well as tilted faces. We tested our method on both the databases. Some of the inaccurate detections of facial feature points are shown in Figure 7.

Table I and Table II shows the detailed analysis of results. By keeping the exact feature point at the center within  $5 \times 5$  window, if point is detected, we consider that point is detected otherwise we consider it to be false detection. Table 1 gives the facial feature point detection rates for samples from Cohn-Kanade database, while Table 2 gives the facial

Table I  
FACIAL FEATURE POINT DETECTION RESULTS FOR SAMPLE FROM COHN  
KANADE DATABASE

Label	Description	Detection rate
A1	Outer corner of left eye	85%
A2	Inner corner of left eye	87%
B1	Outer corner of right eye	83%
B2	Inner corner of right eye	82%
C1	Outer corner of left eyebrow	81%
C2	Inner corner of left eyebrow	80%
D1	Outer corner of right eyebrow	85%
D2	Inner corner of right eyebrow	84%
E1	Eyeball of left eye	87%
E2	Eyeball of right eye	90%
F1	Left nostril	90%
F2	Right nostril	91%
G1	Left mouth corner	92%
G2	Right mouth corner	90%
Average rate for all points		86%

Table II  
FACIAL FEATURE POINT DETECTION RESULTS FOR SAMPLE FROM IMM  
DATABASE

Label	Description	Detection rate
A1	Outer corner of left eye	82%
A2	Inner corner of left eye	83%
B1	Outer corner of right eye	81%
B2	Inner corner of right eye	80%
C1	Outer corner of left eyebrow	81%
C2	Inner corner of left eyebrow	82%
D1	Outer corner of right eyebrow	84%
D2	Inner corner of right eyebrow	81%
E1	Eyeball of left eye	80%
E2	Eyeball of right eye	85%
F1	Left nostril	82%
F2	Right nostril	87%
G1	Left mouth corner	88%
G2	Right mouth corner	84%
Average rate for all points		83%

feature point detection rates for samples from IMM database.

#### IV. CONCLUSIONS

In this paper we present a simple method for automatic detection of 14 facial feature points in image sequence of any expression. When tested on Cohn-Kanade database we achieved 86% average accuracy and on IMM database 83% average accuracy. Computational complexity of the method is very less. The method is applicable for frontal as well as tilted faces, and for face with any expressions or without expressions. Exact feature points are located for many samples, while for some samples nearby points are located. In many computer vision applications approximate locations of feature points is required with less computations. In such situations this technique can be used.

The work is ongoing towards improving the existing accuracy of detection, utilizing Gabor filter of one fixed scale and orientation. Instead of using bank of Gabor filters we try to use single filter with our approach.

#### REFERENCES

- [1] M. Reinders, R. Koch, and J. Gerbrands, "Locating facial features in image sequences using neural networks," in *Proc. IEEE International Conference on Face and Gesture Recognition*, pp. 230–235, Oct 1996.
- [2] E. jung Holden and R. Owens, "Automatic facial point detection," in *Proc. Asian Conf. Computer Vision*, pp. 731–736, 2002.
- [3] R. S. Feris, J. Gemmell, K. Toyama, and V. Krueger, "Hierarchical wavelet networks for facial feature localization," in *Proc. IEEE International Conference on Face and Gesture Recognition*, pp. 118–123, 2002.
- [4] C. Hu, R. Feris, and M. Turk, "Real-time view-based face alignment using active wavelet networks," in *Proc. IEEE International Workshop Analysis and Modeling of Faces and Gestures*, pp. 215–221, 2003.
- [5] S. Yan, X. Hou, S. Z. Li, H. Zhang, and Q. Cheng, "Face alignment using view-based direct appearance models," *International Journal of Imaging Systems and Technology*, vol. 13, pp. 106–112, 2003.
- [6] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 775–779, Jul 1997.
- [7] D. Cristinacce and T. Cootes, "Facial feature detection using AdaBoost with shape constraints," in *14<sup>th</sup> British Machine Vision Conference, Norwich, England*, pp. 231–240, 2003.
- [8] I. Cohen, N. Sebe, A. Garg, L. S. Chen, and T. S. Huang, "Facial expression recognition from video sequences: Temporal and static modeling," *Computer Vision and Image Understanding*, vol. 91, pp. 160–187, 2003.
- [9] D. Vukadinovic and M. Pantic, "Fully automatic facial feature point detection using gabor feature based boosted classifiers," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1692–1698, Oct. 2005.
- [10] W. Kienzle, G. BakIr, M. Franz, and B. Scholkopf, "Face detection - efficient and rank deficient," in *Advances in Neural Information Processing Systems*, vol. 17, pp. 673–680, 2005.
- [11] P. Viola and M. Jones, "Robust real-time object detection," tech. rep., February 2001.
- [12] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.
- [13] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 46–53, 2000.
- [14] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, "FAME – a flexible appearance modelling environment," *IEEE Trans. on Medical Imaging*, vol. 22, no. 10, pp. 1319–1331, 2003.
- [15] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.
- [16] A. Nikolaidis and I. Pitas, "Facial feature extraction and determination of pose," *Pattern Recognition*, vol. 33, pp. 1783–1791, 1998.
- [17] J. Ahlberg, "Facial feature extraction using deformable graphs and statistical pattern matching," in *Swedish Symposium on Image Analysis, SSAB*, 1999.
- [18] I. Kotsia and I. Pitas, "Facial expression recognition in image sequences using geometric deformation features and support vector machines," *IEEE Transactions on Image Processing*, vol. 16, pp. 172–187, Jan. 2007.
- [19] H. Wu, T. Yokoyama, D. Pramadhanto, and M. Yachinda, "Face and facial feature extraction from color image," in *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, pp. 343–350, 1996.
- [20] C. J. Lien, T. Kanade, J. F. Cohn, and C. Li, "Detection, tracking, and classification of action units in facial expressions," *J. Robot Auton. Sys.*, July 1999.
- [21] B. Fasel and J. Luetttin, "Automatic facial expression analysis: A survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.
- [22] G. Edwards, T. Cootes, and C. Taylor, "Face recognition using active appearance models," in *Proc. European Conf. Computer Vision*, vol. 2, pp. 581–695, 1998.