Multi-State Based Facial Feature Tracking and Detection

Ying-li Tian   Takeo Kanade   Jeffrey F. Cohn

CMU-RI-TR-99-18

Robotics Institute, Carnegie Mellon University,
Pittsburgh, PA 15213

August, 1999

This research is sponsored by NIMH, under contract 1R01MH51435 “Facial Expression Analysis by Computer Image Processing”.

Keywords: Multiple-state model Lip tracking Eye tracking Furrow detection

Abstract

Accurately and robustly tracking facial features must cope with the large variation in appearance across subjects and the combination of rigid and non-rigid motion. We present a work toward a robust system to detect and track facial features including both permanent (e.g. mouth, eye, and brow) and transient (e.g. furrows and wrinkles) facial features in a nearly frontal image sequence. Multi-state facial component models are proposed for tracking and modeling different facial features. Based on these multi-state models, and without any artificial enhancement, we detect and track the facial features, including mouth, eyes, brows, cheeks, and their related wrinkles and facial furrows by combining color, shape, edge and motion information. Given the initial location of the facial features in the first frame, the facial features can be detected or tracked in remainder images automatically. Our system is tested on 500 image sequences from the Pittsburgh-Carnegie Mellon University (Pitt-CMU) Facial Expression Action Unit (AU) Coded Database, which includes image sequences from children and adults of European, African, and Asian ancestry. Accurate tracking results are obtained in 98% of image sequences.

1. Introduction

The face is the main source of information for discrimination and identification of people. It is constitutes the structural ground of many nonverbal messages, including information about the emotional state of the person [3]. Facial analysis has received numerous attention in the human face detection, recognition and man-machine interface literature since the early 1970's [2, 5, 10, 12, 14, 30]. Robust and accurate analysis of facial features must cope with the large variation in appearance across subjects and the large appearance variability of a single subject caused by changes in lighting, pose, and facial expressions. Mouth and eye features play a central role for automatic face recognition, facial expression analysis, lip-readings and speech processing. Accurate localization of facial features is both difficult and computationally expensive if performed for each frame. So we manually locate the templates of lips, eyes, eyebrows and cheeks in the first frame and track them through the remainder of the sequence. For the furrow information, we detect them in each frame. Tracking lip and eye motion accurately and robustly in image sequences is especially difficult because these features are highly deformable, vary in shape, color, specularity, and relation to surrounding features across individuals, and are subject to both non-rigid (expression) and rigid motion (i.e., head movement). Although many lip and eye tracking methods have been proposed, they have the limitations for obtaining robust results.

For facial expression analysis [11, 19], more exact extraction of subtle facial features is required. It is very difficult to construct a computer vision algorithm which is flexible enough to cope with the huge variety of human facial appearances. In the field of gesture recognition, state-based models have been used [1, 13, 27]. A gesture is often defined to be a sequence of states in a measurement or configuration space. Transitions can occur between these states. The repeatability and variability of the trajectories through the state space can be measured by training examples. To develop a facial expression analysis system which is robust to rigid motion (head motion) and non-rigid motion (expression), we proposed a multi-state face and facial component model. The basic concept is that there are multiple states for a face based on the head orientation. For each different face state, different facial components are used. For each facial component, there are also several different states. For each different state, different vision algorithm maybe used to obtain the best result.

We develop an accurate and robust system for permanent facial feature (e.g. mouth, eye, brow and cheek) tracking and transient (e.g. furrows and wrinkles) facial feature detection in image sequence based on the multi-state model. We have tested our system in the *Pitt-CMU Facial Expression AU Coded Database* that include more than 5000 images of different kinds of people (including Caucasian, Afro-American, Hispanic and Asian) and expressions. Excellent results have been obtained even when there is head motion.

In Section 2, the multi-state models for face and facial components are introduced. The lip tracking method is described in Section 3. We present the eye tracking method in Section 4. The brow and cheek tracking and the transient facial feature detection are developed in Section 5. Section 6 is the experimental results. Section 7 is the conclusion and discussion.

2. Multi-State Models for Face and Facial Components

2.1. Multi-State Face Model

One of the most significant factors affecting the appearance of a face is the head orientation. Seven head states (left, left-front, front, right-front, right, down, and up) are shown in Figure 1 based on the head orientation. The in-plane head rotations do not affect the head state. The head orientation and face position are detected by the face detection procedure [25, 26]. For the different head states, different facial component models are used. For example, the facial component models for a front face include $FrontLips$, $FrontEyes$ (left and right), $FrontCheeks$(left and right), $NasolabialFurrows$, and $Nosewrinkles$. For the right face, only the component models such as $SideLips$, $Righteye$, $Rightbrow$, and $Rightcheek$ are used. In our current system, we assume the face images are nearly front view including the in plane head rotations (Figure 2).

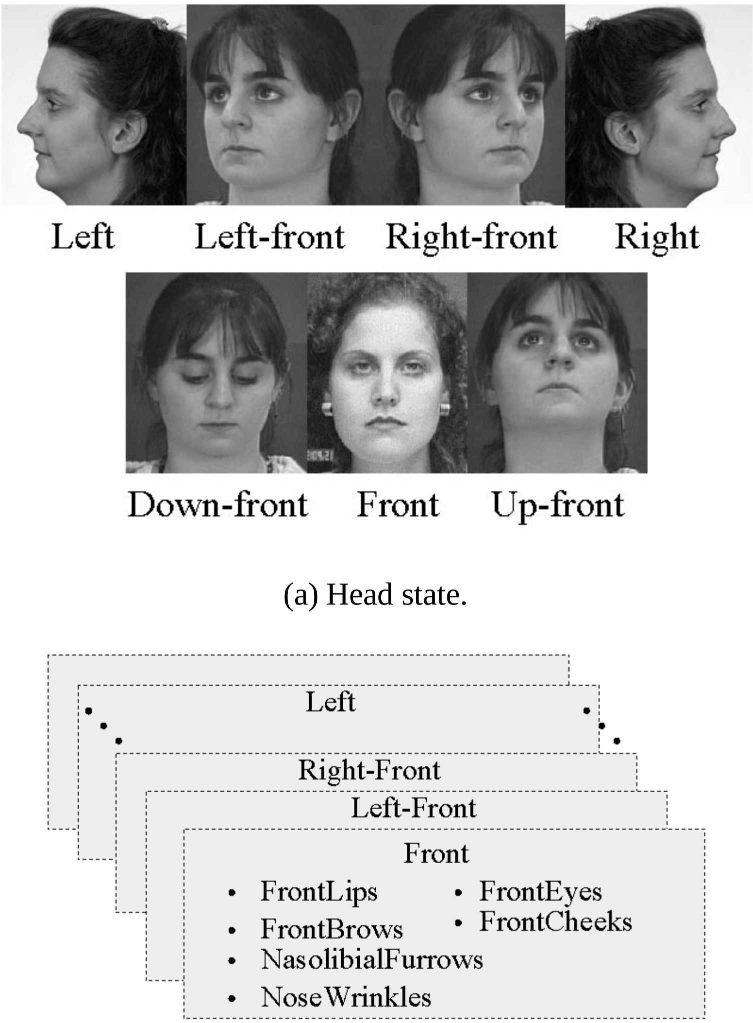
2.2. Multi-State Face Component Models

Different face component models must be used for different states. For example, a lip model of the front face does not work for a profile face. Here, we give the detailed facial component models for the nearly front-view face. Both the permanent components such as lips, eyes, brows, cheeks and the transient components such as furrows are considered. Based on the different appearances of different components, different geometric models are used to model the component's location, shape, and appearance. Each component employs a multi-state model corresponding to different component states. For example, a three-state lip model is defined to describe the lip states: open, closed, and tightly closed. A two-state eye model is used to model open and closed eye. There is one state for brow and cheek. Present and absent are use to model the states of the transient facial features. The multi-state component models for different components are described in Table 1.

3. Lip Tracking

3.1. Lip Tracking Problems

Each of lip tracking methods that have been proposed so far has its own strength and limitations. We believe that a feature extraction system intended to be robust to all the sources of variability (i.e.,



(a) Head state.

(b) Different facial components used for each head state.

Figure 1. Multiple state face model. (a) The head state can be left, left-front, front, right-front, right, down, and up. (b) Different facial component models are used for different head states.

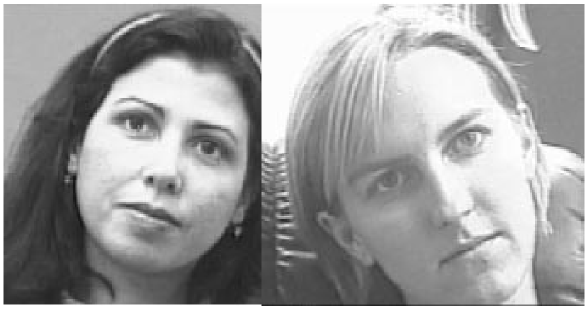
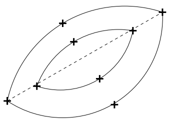
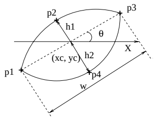
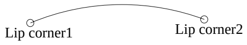
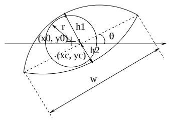
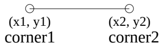
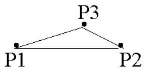
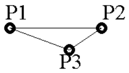
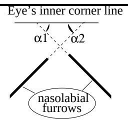
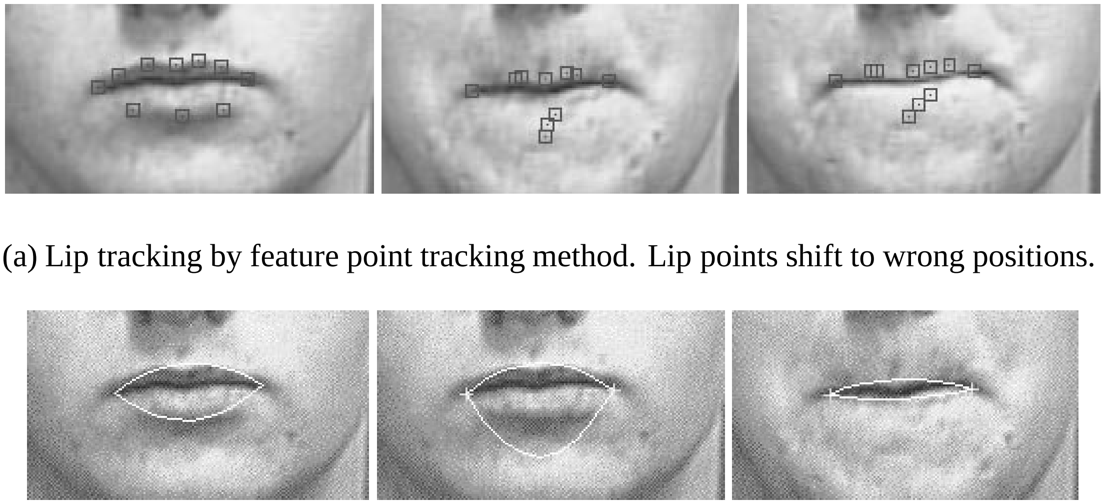


Figure 2. Examples of in-plane head rotation of the front-view face.

Table 1. Multi-state facial component models of a front face

Component	State	Description/Feature
Lip	Opened	
	Closed	
	Tightly closed	
Eye	Open	
	Closed	
Brow	Present	
Cheek	Present	
Furrow	Present	
	Absent	

individual differences in people's appearance, etc.), should use as much knowledge about the scene as possible. Lip tracking methods based on a single cue about the scene are insufficient for tracking lips robustly and accurately. For example, the snake [15] and active contour methods [22] often converge to the wrong result when the lip edges are not clear or lip color is very close to face color. Luettin and Thacker [21] proposed a lip tracking method for speechreading using probabilistic models. Their method needs a large set of training data to learn patterns of typical lip deformation. The lip feature point tracking method of Lien [19] is sensitive to the initial feature points position, and the lip feature points have ambiguity along the lip edges. A feature extraction system should use all available information about the scene to handle all the sources of variability in real environments (illumination, individual appearance, etc.).



(a) Lip tracking by feature point tracking method. Lip points shift to wrong positions.

(b) Lip tracking by color-based deformable template method. Lower lip contour jump to wrong position because the shadow.

Figure 3. Tracking problems by different algorithms. The lip contour or lip points shift to wrong positions.

Many researchers try to combine more information to develop lip trackers. Bregler and his colleagues [7] developed an audio-visual speech recognition system that uses Kass's snake approach with shape constraints imposed on possible contour deformations. They found that the outer lip contour was not sufficiently distinctive. This method uses image forces consisting of gray-level gradients, which are known to be inadequate for identifying the outer lip contour [30]. Yuille *et al.* used the edge, peak and valley information with mouth template for lip location, but there were still some problems during energy minimizing. The weights for each energy term were adjusted by preliminary experiments. This process was time-consuming so can not be used as tracking, and the weights were not applicable to the novel subjects. The color based deformable template method developed by Rao [24] combines shape and color information, but it has difficulty when there is shadow near the lip or the lip color is similar to the face. Examples of some of these problems are shown in Figure 3. The limitations of these methods can be observed clearly. Figure 3(a) shows failure of Lien's feature points tracking when the lip contour becomes occluded. The feature points on the lip shift to wrong positions when lips are tightly closed. Figure 3 (b) shows failure of Rao's color-based deformable method due to shadow and to occlusion. The lower lip contour jumps to chin because the shadows near the bottom lip.

3.2. Multi-State Mouth Model

As shown in Figure 4, we classify the mouth states as open, relatively closed, and tightly closed. We define the lip state as tightly closed if the lips are invisible because of lip suck. For the different states, different lip templates are used to obtain the lip contour (Figure 4 (e), (f), and (g)). For the open mouth, a more complex template could be used that includes inner lip contour and visibility of teeth or tongue. Currently, only the outer lip contour is considered. For the relatively closed mouth, the outer contour of the lips is modeled by two parabolic arcs with six parameters: lip center (xc, yc), lip shape ($h1$, $h2$ and w), and lip rotation (θ). For the tightly closed mouth, the dark mouth line ended at lip corners is used (Figure 4(g)). The state transitions are determined by the lip shape and color.

3.3. Lip Color Distribution

We model the color distribution inside the closed mouth as a Gaussian mixture. There are three prominent color regions inside the mouth: a dark aperture, pink lips, and bright specularity. The density functions of the mouth Gaussian mixtures are given by:

$$f_{mouth}(x) = \sum_{j=1}^{3} w_j N(x|\mu_j, \mathbf{C}_j), \tag{1}$$

where

$$N(x|\mu_j, \mathbf{C}_j) = \frac{1}{(2\pi)^{N/2} | \mathbf{C}_j |^{1/2}}$$
$$exp\{-\frac{1}{2}(x-\mu_j)^T \mathbf{C}_j^{-1}(x-\mu_j)\}. \tag{2}$$

$\{w_j\}$ are the mixture weights($\sum_j^3 w_j = 1, w_j >= 0$), $\{\mu_j\}$ are the mixture means, and $\{\mathbf{C}_j\}$ are the mixture covariance matrices.

In order to identify the model parameter values, the lip region is manually specified in the first frame image. The Expectation-Maximization (EM) algorithm [8] is used to estimate both the mixture weights

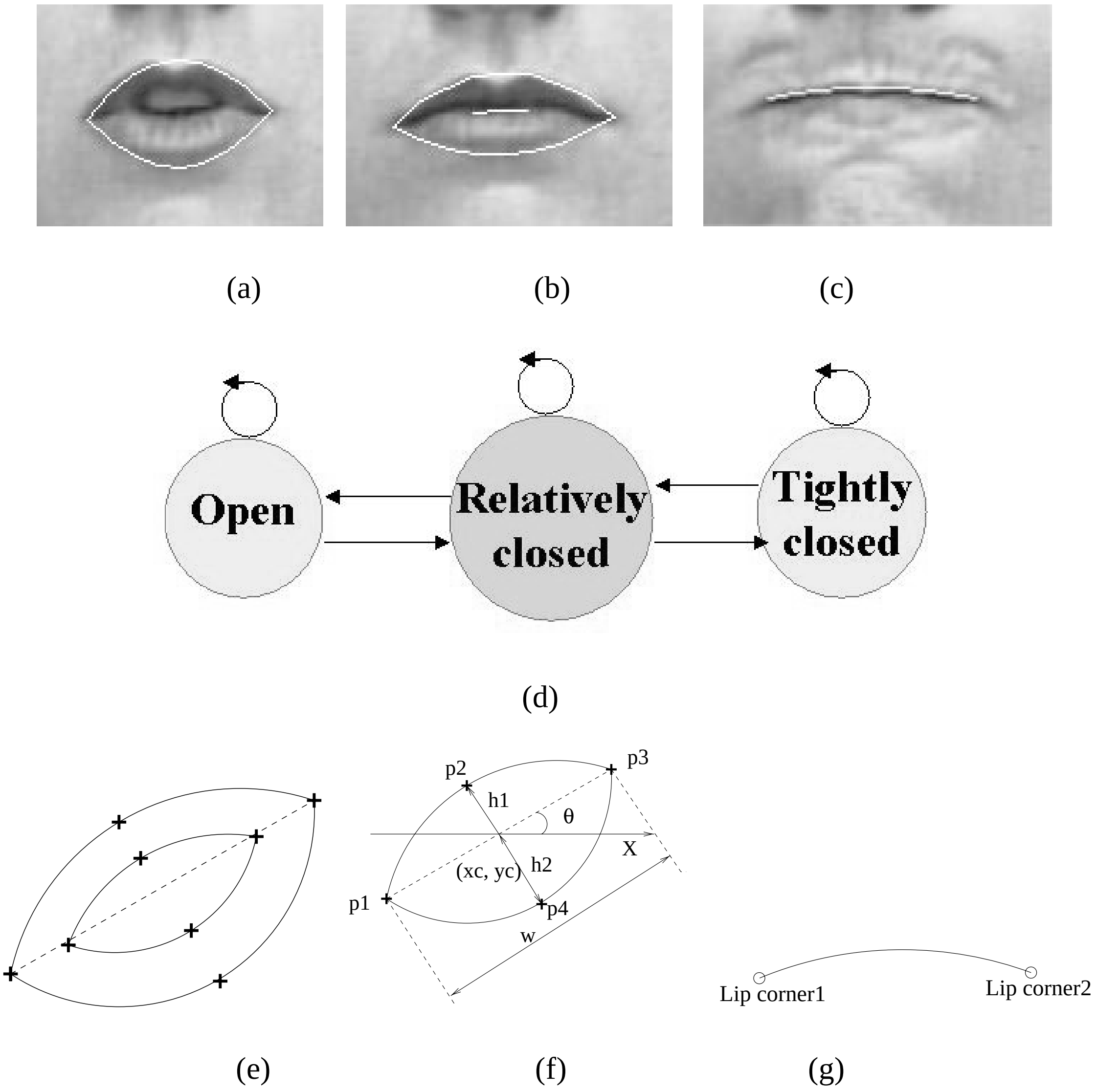


Figure 4. Multi-state mouth model and lip templates. (a) An open mouth. (b) A closed mouth. (c) A tightly closed mouth. (d) The state transition diagram. (e) The open mouth parameter model. (f) The closed mouth parameter model. (g) The tightly closed mouth parameter model.

and the underlying Gaussian parameters. K-means clustering is used to provide initial estimates of the parameters. Once a model is built, the succeeding frames are tested by a look-up table obtained from the first frame.

3.4. Lip Tracking by Combining Shape and Motion Information

Lip motion: In our system, the lip motion is obtained by a modified version of the Lucas-Kanade tracking algorithm [20]. We assume that intensity values of any given region (feature window size) do not change, but merely shift from one position to another. Consider an intensity feature template $I_t(x)$ over a $n \times n$ region R in the reference image at time t; we wish to find the translation d of this region in the following frame $I_{t+1}(x + d)$ at time $t + 1$, by minimizing a cost function E defined as:

$$E = \sum_{x \in R} [I_{t+1}(x + d) - I_t(x)]^2. \tag{3}$$

and the minimization for finding the translation d can be done in iterations:

$$d_{n+1} = d_n + \left\{ \sum_{x \in R} (\frac{\partial I}{\partial x})^T |_{x+d_n} [I_t(x) - I_{t+1}(x)] \right\} \left[\sum_{x \in R} (\frac{\partial I}{\partial x})(\frac{\partial I}{\partial x})^T |_{x+d_n} \right]^{-1}, \tag{4}$$

here d_0, the initial estimate, can be taken as zero if only small displacements are involved.

Consecutive frames of an image sequence may contain large feature-point motion such as sudden head movements, brows raised or mouth opening with a the surprised expression; any or all of these may cause missing or lost tracking. In order to track these large motions without losing sub-pixel accuracy, a pyramid method with reduced resolution is used [23]. Each image is decomposed into 5 levels, from level 0 (the original finest resolution image) to level 4 (the coarsest resolution image). In our implementation, a 5x5 Gaussian filter is used to smooth out the noise in order to enhance the computation convergence, and a 13x13 feature region is used for all levels. The rapid and large displacements of up to 100 pixels can be tracked robustly while maintaining sensitivity to sub-pixel facial motion.

Lip template: A lip template is used to obtain the correct lip region. After locating the lip template in the first frame, only four key points of the lip template are tracked (p1, p2, p3 and p4 in Figure 4(f)) in the remainder of the sequence. The lip corners are tracked exactly and the top lip and bottom lip heights are obtained. From the lip corner positions and the lip heights, the lip contour is obtained by calculating the corresponding lip template parameters. The combined method is more robust and accurate for most images, but fails for the tightly closed lip (Figure 7(a)). We solve this problem by combining color information based on a multi-state mouth model.

3.5. Lip State by Color and Shape

Each lip state and its color distribution is shown in Figure 5. For the open mouth and the tightly closed mouth, there are non-lip pixels inside the lip contours. Assume the lip state in the first frame is neutral closed. From the lip color distribution, we get the lip states by:

$$Lipstate = \begin{cases} Open & \text{if } (h - h_0)/h_0 > 0 \text{ and } \gamma > T_1 \\ Tightly\ closed & \text{if } (h - h_0)/h_0 < 0 \text{ and } \gamma > T_2, \\ Closed & \text{otherwise} \end{cases} \tag{5}$$

where h_0 and h are the sum of top lip and bottom lip heights in the first frame and current frame; $\gamma = n_{nonlip}/n$, n_{nonlip} is non-lip pixels number inside lip contour and n is all pixels number inside lip contour; $T_1 = 0.35$ and $T_2 = 0.25$ are thresholds in our application.

3.6. Lip Contour for Tightly Closed Lip

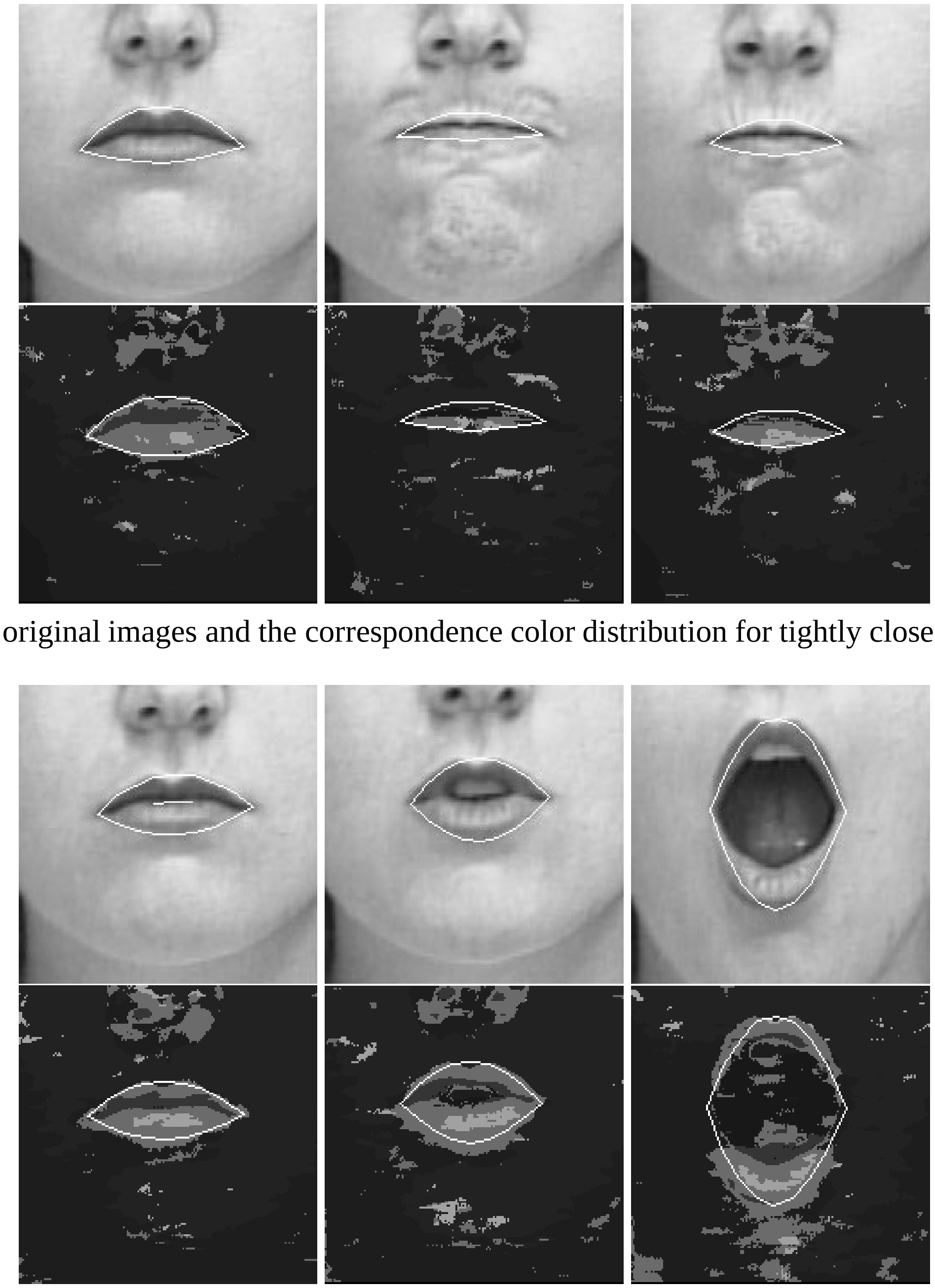
For the tightly closed mouth, we trace the mouth line by locating the darkest pixels along the perpendicular lines with a distance r between two lip corners(Figure 6).

Based on the multi-state mouth model, the color, shape, and motion information are combined in our lip tracking method. This method is very robust to tracking lip for each state. Figure 7(b) shows that the lip contours are tracked correctly by our method for the tightly closed lip.

4. Eye Tracking

4.1. Eye Tracking Problems

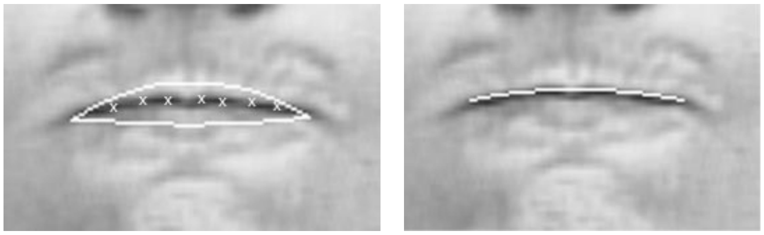
Eye tracking has received a great deal of attention. However, most eye trackers only work well for open eyes and simply track the locations of the eyes. Blinking is a physiological necessity for humans. Moreover, for applications such as facial expression analysis and driver awareness systems, we need to do more than simply track the locations of the person's eyes, but also obtain a detailed description of the eye. We need to recover the state of the eyes (i.e. whether they are open or closed), the parameters of an eye model (e.g. the location and radius of the iris, and the corners and height of the eye opening).



(a) The original images and the correspondence color distribution for tightly closed mouth

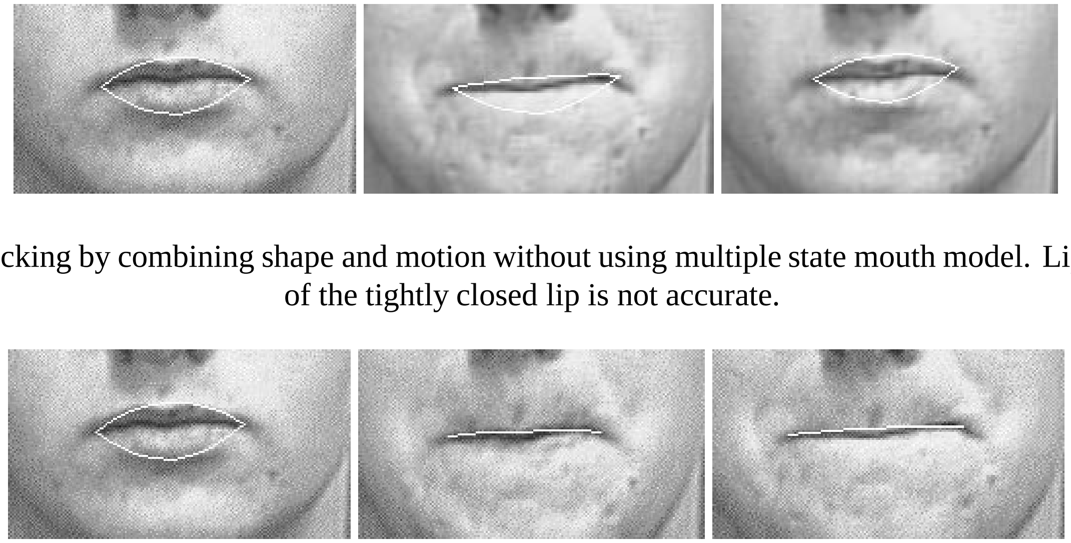
(b) The original images and the correspondence color distribution open mouth

Figure 5. Get Lip states from lip color information. For the open mouth and the tightly closed mouth, there are non-lip pixels inside the lip contours.



(a) Approximate contour (b) Refined contour

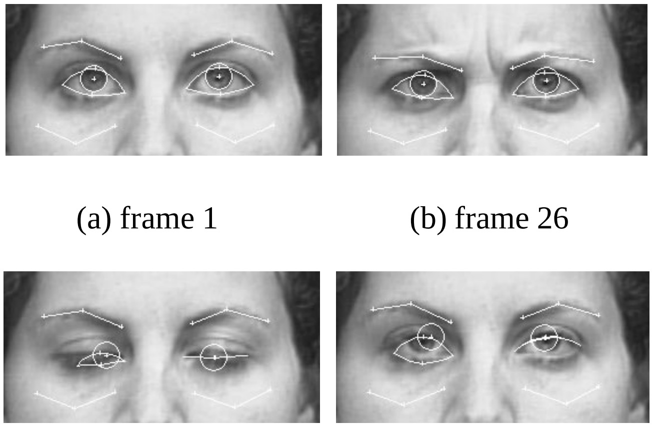
Figure 6. Get tightly closed lip line.



(a) Lip tracking by combining shape and motion without using multiple state mouth model. Lip contour of the tightly closed lip is not accurate.

(b) Lip tracking by combining shape, color and motion based on multiple state mouth model. Lip contours are obtained accurately.

Figure 7. Tracking results by OUR METHOD for the same image sequence in Figure 3.



(a) frame 1 (b) frame 26

(c) frame 32 (d) frame 39

Figure 8. Eye tracking using feature point tracking [10] in the presence of eye closure. As the eyes close, the feature points disappear and so the tracker fails. After the eye opens, the iris in the left eye and the contour of the right eye have been lost.

Tracking eye parameters and detecting eye states is more difficult than just tracking eye locations because the eyes occupy a small region of the face, there is little color information or intensity contrast, and because eye blinking and winking is distracting. Eye feature extraction from static images was developed by Kanade and other researchers [5, 9, 12, 14, 18, 29, 30]. Most eye feature extraction methods have been improved based on Yuille's deformable template method to track both eye locations and extract their parameters [5, 9, 18, 29, 30]. However, the deformable template scheme is time consuming and hard be used as an eye tracker. Also the template must be started at or below the eye, otherwise, it will locate the eyebrow instead of the eye. Chow et al. [5] used a two-step approach based on the Hough transform and then the deformable template to extract the features. The Hough transform is used to locate the approximate position of the iris. An improved method of using deformable templates was introduced by Xie *et al.* [29]. Lam and Yan [18] proposed an eye-corner based deformable template method for locating and extracting eye features. But it was still too slow to use for tracking. Deng [9] *et al.* proposed a region-based deformable template method for locating the eyes and extracting eye model parameters. They tried to use their method to track upper eyelid movement when eye position and eye size are almost fixed in an image sequence.

Previous systems [19] have used feature point tracking to track the eyelid, however such an approach is prone to error if the eyes blink in the image sequence. An example of the kind of mis-tracking that can occur is shown in Figure 8. As the eyelid closes the feature points on the eye contour disappear momentarily, but yet long enough for the tracker to loose them. After the eye opens the iris in the left eye and the contour in the right eye have been completely lost.

4.2. Dual-State Eye Model

As shown in Figure 9, we define an eye state is open or closed in the image sequences. The iris can provide important information about the eye state because if the eye is closed the iris will not be visible, but if the eye is open part of the iris will normally be visible. If the iris is detected, the eye is open. Otherwise, the eye is closed. For the different states, different eye templates and different algorithms are used to obtain eye features.

For an open eye, we use the same eye template as Yuille's except for two points located at the center of the whites [30]. The template, illustrated in Figure 9 (d), is composed of a circle with three parameters (x_0, y_0, r) and two parabolic arcs with six other parameters $(x_c, y_c, h_1, h_2, w, \theta)$. We assume the outer contour of the eye is symmetrical about the perpendicular bisector to the line connecting two eye corners. For a closed eye we used a straight line for the template. It has 4 parameters, two for each of the two end-points. This template, illustrated in Figure 9 (e), is sufficient for describing closed eye features.

4.3. Eye Position Initialization

We assume the initial location of the eye is given in the first frame. The purpose of this stage is to get the initial eye position in the first frame of the image sequence. Some literature about eye locating has been published [5, 9, 18, 29, 30].

4.4. Eye Region Intensity Normalization

For some image sequences, the eye region is very dark because of eye makeup or poor illumination. In this case, our tracker can not track the correct eye boundary. We therefore normalize the intensity of the image sequence. After the eye positions are initialized, a fixed size window is taken around the eye region. The intensities in this region are linearly stretched to fill the $0 - 255$ range. For color

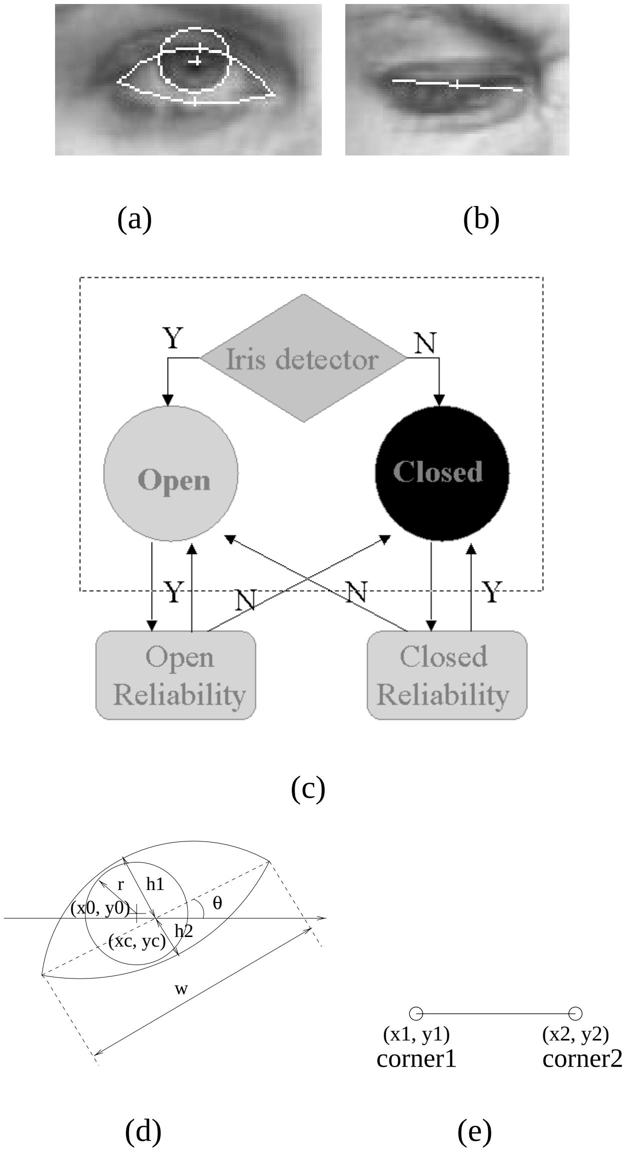
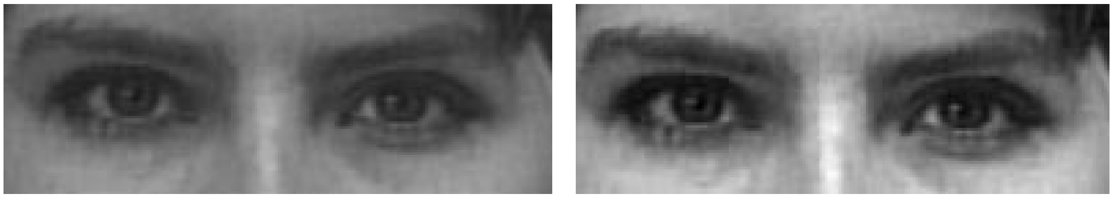


Figure 9. Dual-state eye model. (a) An open eye. (b) A closed eye. (c) The state transition diagram. (d) The open eye parameter model. (e) The closed eye parameter model.

image sequences, the R, G, B channels are stretched separately. Figure 10 shows the original eye image and the normalized image. In experiments, we found that our tracker works well after this intensity normalization.



(a) Original image (b) Normalized image

Figure 10. Eye region intensity normalization.

4.5. Eye Corner Tracking

4.5.1. Inner Corners

We found that eye inner corners are the most stable features in a face and relatively insensitive to facial expressions. Using an edge-based corner detector, the inner corners can be detected easily. However, due to the low intensity contrast at the eye boundary and the wrinkles around the eye, some false corners will be detected as well as the true corners. Instead of using the corner matching method, we therefore use a feature point tracking method to track the eye inner corners for the remaining images of the sequence. In our system, the eye inner corners are tracked by a modified version of the Lucas-Kanade tracking algorithm [20].

4.5.2. Outer Corners

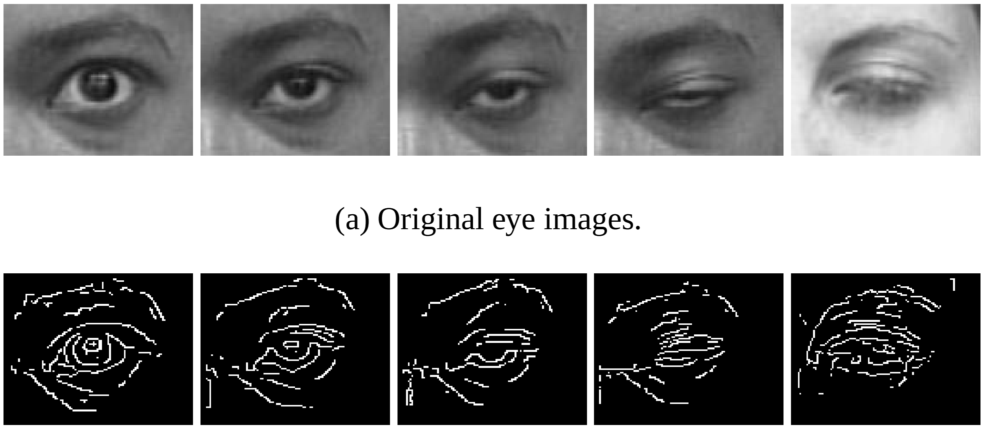
We found the outer corners of the eyes are hard to detect and less stable than the inner corners. We assume the outer corners are collinear with the inner corners. The eye shape information (width of the eye and distance between two inner corners) is obtained from the first frame. After tracking the inner corners in each frame, the positions of the outer corners can be obtained from eye shape information.

4.6. Iris Tracking and Eye State Detection

4.6.1. Iris Mask

The iris can provide important information about the eye state because if the eye is closed the iris will not be visible, but if the eye is open part of the iris will normally be visible. Generally, the pupil is the darkest area in the eye region. Some eye trackers locate the iris center by searching for the darkest area in each frame. However, the iris center often shifts to the eye corner, dark eyelash, or eyelids with eye shadow. In our system, we use the iris intensity and edge map to track the iris center.

We use a Canny edge operator [4] to get the eye edge maps. The edge detection results for several different open eye stages are shown in Figure 11. We found that the edge maps are very noisy even in a clear eye image, so we do not use the edge map directly. We observed that the iris edge is relative clear, and the upper part of the iris is often occluded by the upper eyelid. As a result, we use a half circle mask to filter the iris edge (Figure 12). The radius of the iris circle template r_0 can be obtained from the first frame. In face image sequences, if there is no large off plane movement of the head, then the iris radius will not change much. We increase and decrease the radius of the circle a little (δr) from r_0 to generate the half circle mask with minimum radius $(r_0 - \delta r)$ and maximum radius $(r_0 + \delta r)$. If the thickness of the iris mask (δr) is too large, many non-iris edges will be included. If δr is too small, the iris edge will not be selected when there is head motion. In our system, we let δr be $r_0/3$.



(a) Original eye images.

(b) Eye edge maps.

Figure 11. Eye edge maps for eyes from wide open to closed. The edge of lower part of the iris is relative clear for an open eye.

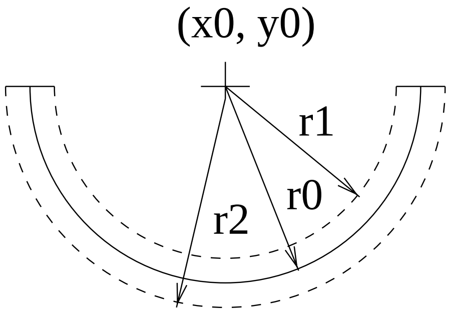


Figure 12. Half circle iris mask. (x_0, y_0) is the iris center; r_0 is the iris radius; r_1 is the minimum radius of the mask; r_2 is the maximum radius of the mask

4.6.2. Iris Tracking and Eye State Detection

Intensity and edge information are used to detect an iris. If the iris is detected, the eye is open and the iris mask center (x_0, y_0) is the iris center. The iris center can be obtained in four steps:

1. Calculate the average intensity I_0 of the lower half of the iris in the first frame.
2. Extract the edge maps in the eye region and calculate the number of pixels belong to edges E_0 in the area between r_1 and r_2 of the iris mask in the first frame.
3. Search the eye region between the inner corner and the outer corner to find the iris mask center (x_0, y_0) with the largest edge pixel number E.
4. Calculate the average intensity I of the iris mask when it is in the position with largest edges. If $(I - I_0) < T_1$ and $E/E_0 > T_2$, the iris is detected with the center (x_0, y_0), where T_1 and T_2 are the thresholds of the intensity and edge respectively. In our system, $T_1 = 30$ and $T_2 = 0.35$.

Eye state can be determined by equation (6).

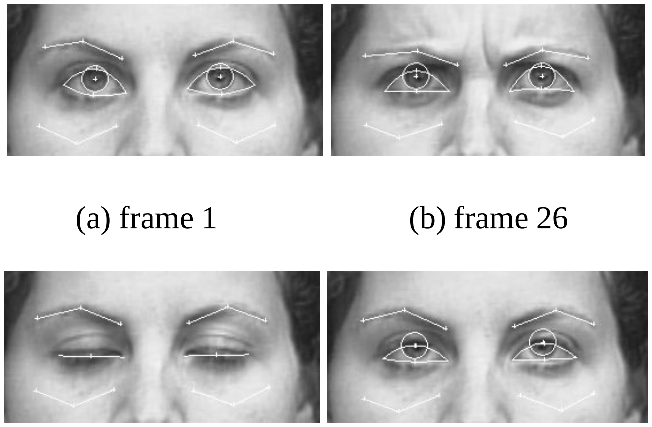
$$Eyestate = \begin{cases} Open & \text{if the iris detected} \\ Closed & \text{otherwise} \end{cases} \quad (6)$$

4.7. Eye Boundary Tracking

For open eye boundary tracking, the eye template is used to obtain the correct eye boundaries. After locating the eye template in the first frame, only two key points of the eye template are tracked (the center points of the upper and the lower eyelids) in the remaining open eye images. From these two points and the eye corners, the eye template parameters can be obtained. Then the eye boundaries are calculated from the corresponding eye template parameters.

For a closed eye, a simple line between the inner and outer eye corners is used as the eye boundary.

Compared to Figure 8, the eye tracking results by our method for images in the sequence after the eye is closed are given in Figure 13. The iris and eye contours are tracked correctly by our method.



(a) frame 1

(b) frame 26

(c) frame 32

(d) frame 39

Figure 13. Eye tracking results by OUR METHOD for image sequence with closed eyes. As an eye is closed, the eye position shown as a simple line connected two corners of the eye. After the eye is open, eyelid boundaries and iris are correctly tracked.

5. Brow, Cheek Tracking and Transient Features Detection

5.1. Brow and Cheek Tracking

Features in the brow and cheek areas are also important to facial expression analysis. One state is used for the brow and cheek. A triangular template with six parameters $(x1, y1)$, $(x2, y2)$, and $(x3, y3)$ is used to model the position of brow or cheek. Both brow and cheek are tracked by feature point tracking.

5.2. Transient Features Detection

Facial motion produces transient features. Wrinkles and furrows appear perpendicular to the motion direction of the activated muscle. These transient features provide crucial information for the recognition of action units. Contraction of the corrugator muscle, for instance, produces vertical furrows between the brows, which is coded in FACS as AU 4, while contraction of the medial portion of the frontalis muscle (AU 1) causes horizontal wrinkling in the center of the forehead.

Some of these lines and furrows may become permanent with age. Permanent crows-feet wrinkles around the outside corners of the eyes, which is characteristic of AU 6 when transient, are common in adults but not in infants. When lines and furrows become permanent facial features, contraction of the corresponding muscles produces changes in their appearance, such as deepening or lengthening. The presence or absence of the furrows in a face image can be determined by geometric feature analysis [19, 17], or by eigen-analysis [16, 28]. Kwon and Lobo [17] detect furrows by snake to classify pictures of people into different age groups. Lien [19] detected whole face horizontal, vertical and diagonal edges for face expression recognition.

In our system, we currently detect nasolabial furrows, nose wrinkles, and crows feet wrinkles. We define them in two states: present and absent. Compared to the neutral frame, the wrinkle state is present if the wrinkles appear, deepen, or lengthen. Otherwise, it is absent. After obtaining the permanent facial features, the areas with furrows related to different AUs can be decided by the permanent facial feature locations. We define the nasolabial furrow area as the area between eye's inner corners line and lip corners line. The nose wrinkle area is a square between two eye inner corners. The crows feet wrinkle areas are beside the eye outer corners.

We use canny edge detector to detect the edge information in these areas. For nose wrinkles and crows feet wrinkles, we compare the edge pixel numbers E of current frame with the edge pixel numbers E_0 of the first frame in the wrinkle areas. If E/E_0 large than the threshold T, the furrows are present. Otherwise, the furrows are absent. For the nasolabial furrows, we detect the continued diagonal edges. The nasolabial furrow detection results are shown in Figure 14.

6. Experimental Results

Image data were 100 image sequences(approximately 5000 images) from the Pitt-CMU Facial Expression AU Coded Database. Subjects ranged in age from 3 years to 30 years and included males and females of European, African, and Asian ancestry. They were videotaped in an indoor environment with

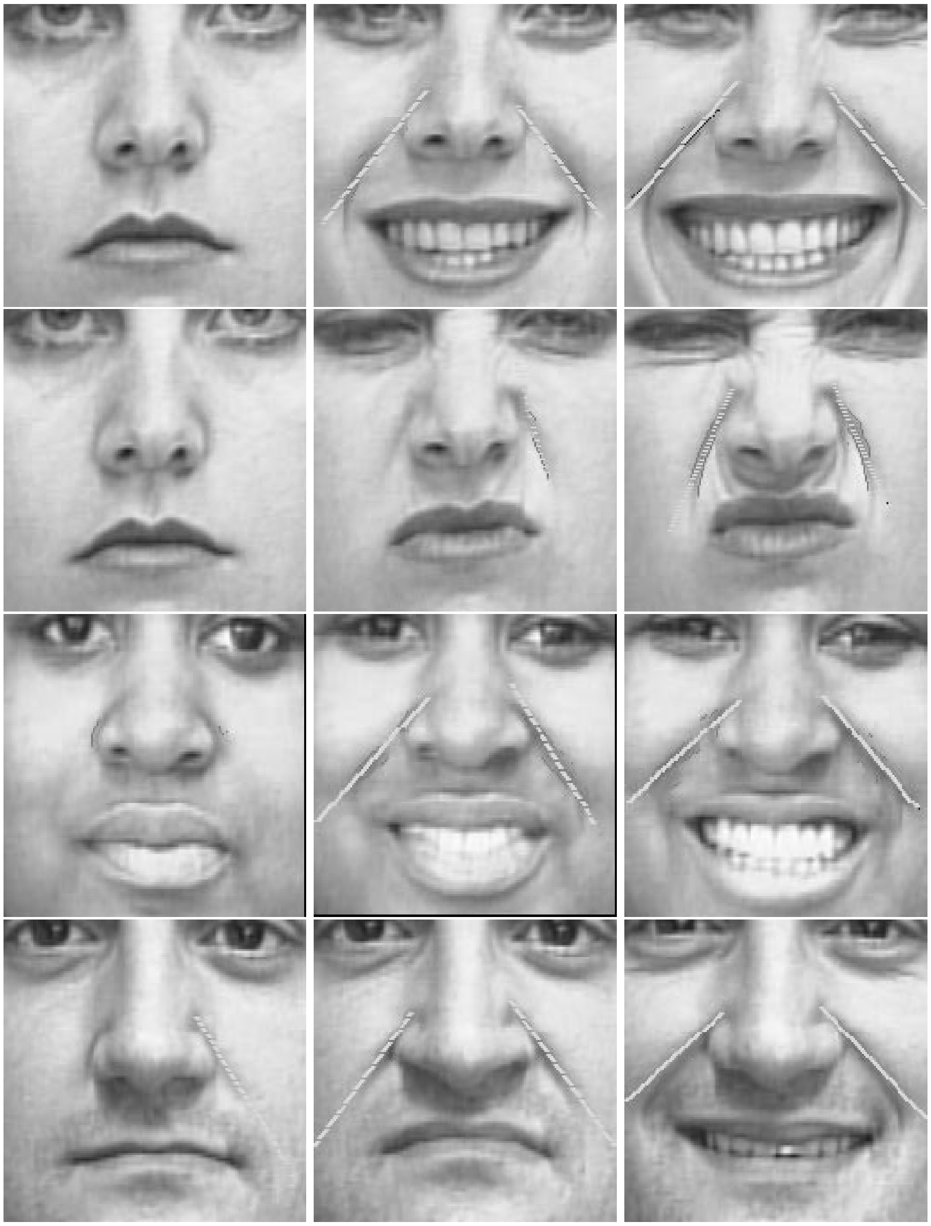


Figure 14. Nasolabial furrow detection results. For the same subject, the nasolabial furrow angle(between the nasolabial furrow and the line connected eye inner corners) is different for different expressions.

uniform lighting. During recording, the camera was positioned in front of the subjects and provided for a full-face view. Images were digitized into 640x480 pixel arrays with 24-bit color resolution. A large variety of facial expression was represented [6].

The permanent feature tracking results for different subjects and different expressions are shown in Figure 15, Figure 16, and Figure 17. Figure 15 shows tracking results for same subject with different expressions. From the happy, surprise and anger expression sequences, we can see that lip states transit each other. From the anger, fear, and disgust, the correct iris position and eye boundaries were tracked after the eye was tightly closed or blink. Notice the semi-circular iris model accurately tracks the iris even when it is only partially visible. For dark skin subjects, good tracking results are obtained also (Figure 16). The robustness of our algorithm in the performance of non-rigid, rigid, and background motion are demonstrated in Figure 17. We test our method in 500 image sequences, the correct tracking ratio is 98%.

7. Conclusion and Discussion

We have described a robust system for tracking facial features include both permanent features (e.g. mouth, eye, and brow) and transient features (e.g. furrows and wrinkles)in a nearly frontal image sequence. To detect qualitative changes of facial features, we develop a multi-state model based system that uses convergent methods of feature analysis. We define the different head orientations and different component appearances as different states. For different head states, different face components are used. For each face component, there are different states also. For each different state, a description and extraction method should be different. Based on these multi-state models, and without artificial enhancement, we detected and tracked mouth, eyes, brow, cheeks, and their related wrinkles and facial furrows. Our system works well regardless of the different subjects and different facial component states.

A limitation of our method is that we assume that the lip template is symmetrical about the perpendicular bisector to the line connecting the lip corners. For non-symmetrical expressions and complex lip shape, there are some errors between the tracking lip contour and the real lip shape (Figure 18). More complex lip template will be necessary to get more accurate lip contours for non-symmetrical expression analysis in our future work. In eye tracking, for a very narrow open eye, for example, Figure 19, sometimes it is detected as closed because too small part of the iris to be detected. In more difficult situations where eyes are covered by hair or eye glasses, environment lighting is changed, or head move to near profile face, current lip and eye template can not provide a practical match. Future work will need to develop different methods for different face states.

Acknowledgements

The authors would like to thank Zara Ambadar and Bethany Peters for testing the method on the database. This work is supported by NIMH grant R01 MH51435.

References

[1] A. Bobick and A. D. Wilson. A state-based technique for the summarization and recognition of gesture. In *International Conference on Computer Vision*, pages 382–388, 1995.

[2] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, Oct. 1993.

[3] E. by R. Bruyer. *The Neuropsychology of Face Perception and Facial Expression*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1986.

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis Mach. Intell.*, 8(6), 1986.

[5] G. Chow and X. Li. Towards a system for automatic facial feature detection. *Pattern Recognition*, 26(12):1739–1755, 1993.

[6] J. F. Cohn, A. J. Zlochower, J. Lien, and T. Kanade. Automated face analysis by feature point tracking has high concurrent validity with manual facs coding. *Psychophysiology*, 36:35–43, 1999.

[7] J. D. Cowan, G. Tesauro, and J. Alspector(eds). *Surface Learning with Applications to Lipreading.* Advances in Neural Information Processing Systems 6, Morgan Kaufmann Publishers, San Francisco, 1994.

[8] A. Dempster, M. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc.,*, (B(39)):1–38, 1977.

[9] J. Deng and F. Lai. Region-based template deformation and masking for eye-feature extraction and description. *Pattern Recognition*, 30(3):403–419, 1997.

[10] P. Ekman, T. Huang, T. Sejnowski, and J. Hager. *NSF Planing Workshop on Facial Expression Understanding*. Arlington, VA, 1992.

[11] I. A. Essa. *Analysis, Interpretation and Synthesis of Facial Expressions*. PHD Thesis, MIT Media Laboratory, 1995.

[12] L. Huang and C. W. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition,* 25(12):1435–1444, 1992.

[13] M. Isard and A. Blake. A mixed-state condensation tracker with automatic model-switching. In *International Conference on Computer Vision,* pages 107–112, 1998.

[14] T. Kanade. *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, Dept. of Information Science, Kyoto University, 1973.

[15] M. Kass, A. Witkin, and D. Terzopoulus. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[16] M. Kirby and L. Sirovich. Application of the k-l procedure for the characterization of human faces. *IEEE Transc. On Pattern Analysis and Machine Intelligence*, 12(1):103–108, Jan. 1990.

[17] Y. Kwon and N. Lobo. Age classification from facial images. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pages 762–767, 1994.

[18] K. Lam and H. Yan. Locating and extracting the eye in human face images. *Pattern Recognition*, 29(5):771–779, 1996.

[19] J.-J. J. Lien, T. Kanade, J. F. Chon, and C. C. Li. Detection, tracking, and classification of action units in facial expression. *Journal of Robotics and Autonomous System*, in press.

[20] B. Lucas and T. Kanade. An interative image registration technique with an application in stereo vision. In *The 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[21] J. Luettin and N. A. Tracker. Speechreading using probabilistic models. *Computer vision and Image Understanding*, 65(2):163–178, Feb. 1997.

[22] J. Luettin, N. A. Tracker, and S. W. Beet. *Active Shape Models for Visual Speech Feature Extraction*. Electronic Systems Group Report No. 95/44, University of Sheffield, UK, 1995.

[23] C. Poelman. The paraperspective and projective factorization method for recovering shape and motion. *Technical Report CMU-CS-95-173, Carnegie Mellon University*, 1995.

[24] R. R. Rao. *Audio-Visal Interaction in Multimedia*. PHD Thesis, Electrical Engineering, Georgia Institute of Technology, 1998.

[25] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, June,1998.

[26] H. Schneiderman and T. Kanade. probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98)*, pages 45–51, July,1998.

[27] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *International Workshop on Automatic Face and Gesture Recognition*, pages 185–194, Zurich, 1995.

[28] M. Turk and A. Pentland. face recognition using eigenfaces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[29] X. Xie, R. Sudhakar, and H. Zhuang. On improving eye feature extraction using deformable templates. *Pattern Recognition*, 27(6):791–799, 1994.

[30] A. Yuille, P. Haallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision,*, 8(2):99–111, 1992.

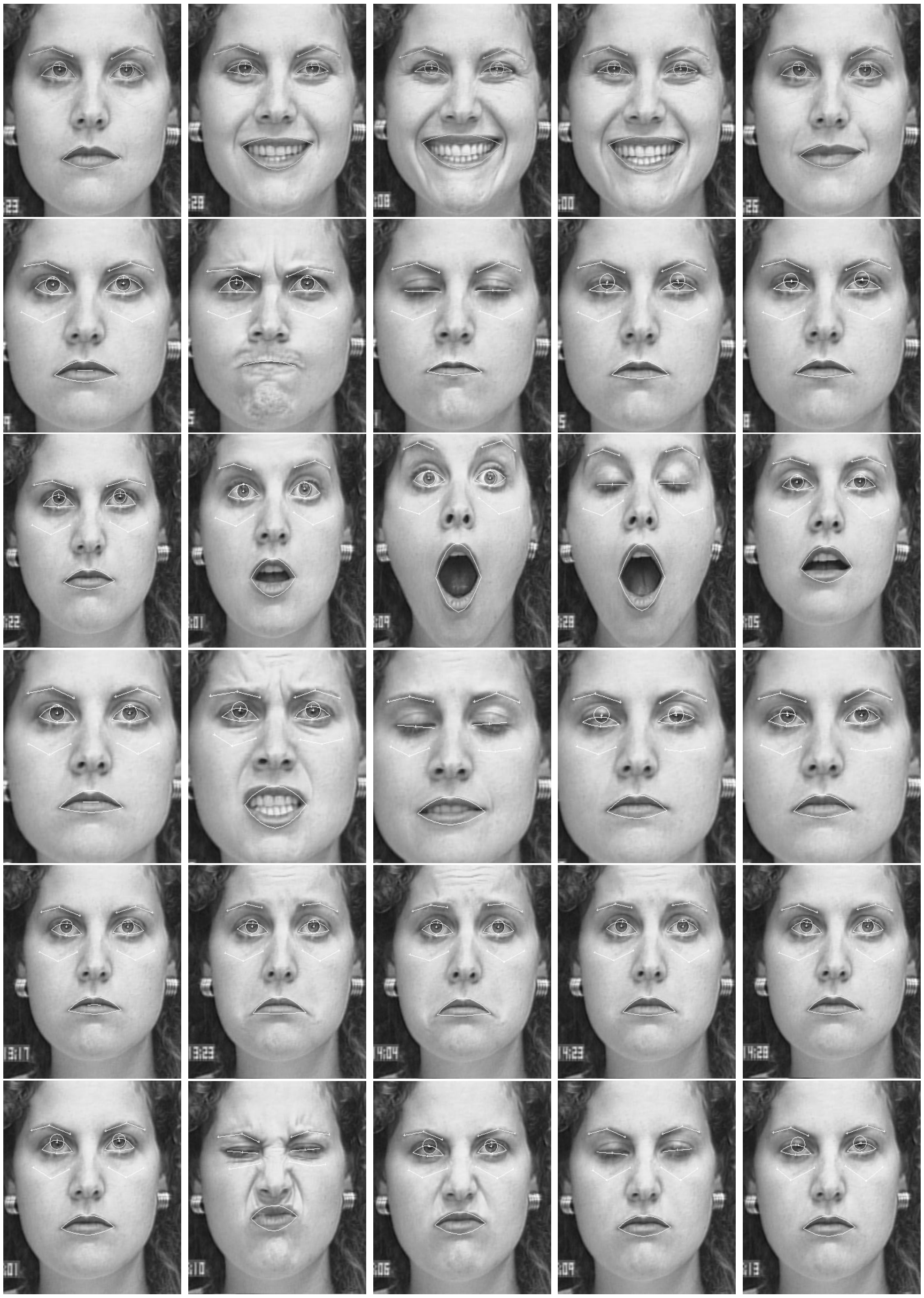


Figure 15. Tracking results by our method for different expressions of the same subject(happy, anger, surprise, fear, sad, and disgust).

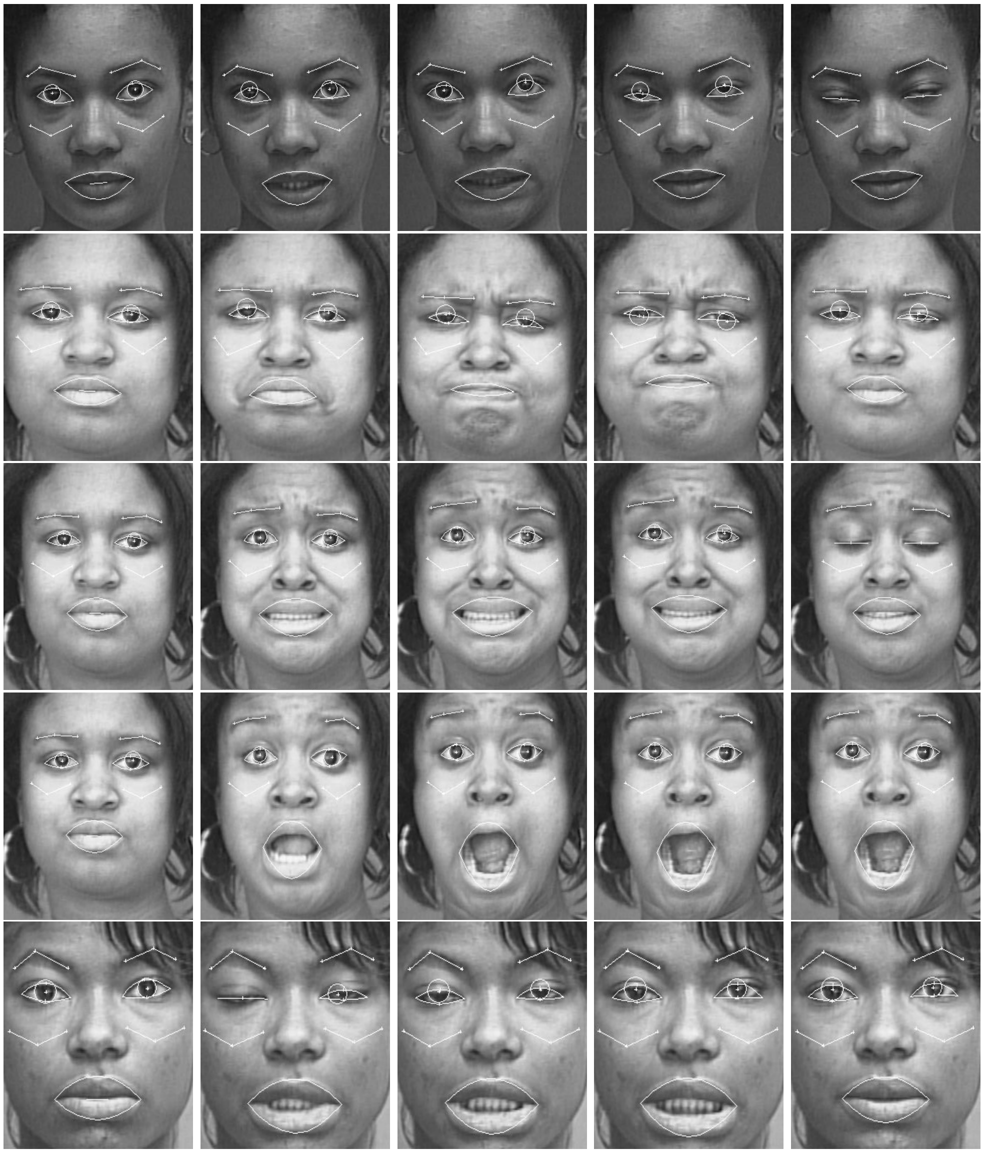


Figure 16. Tracking results by our method for darkskin subjects with different expressions.

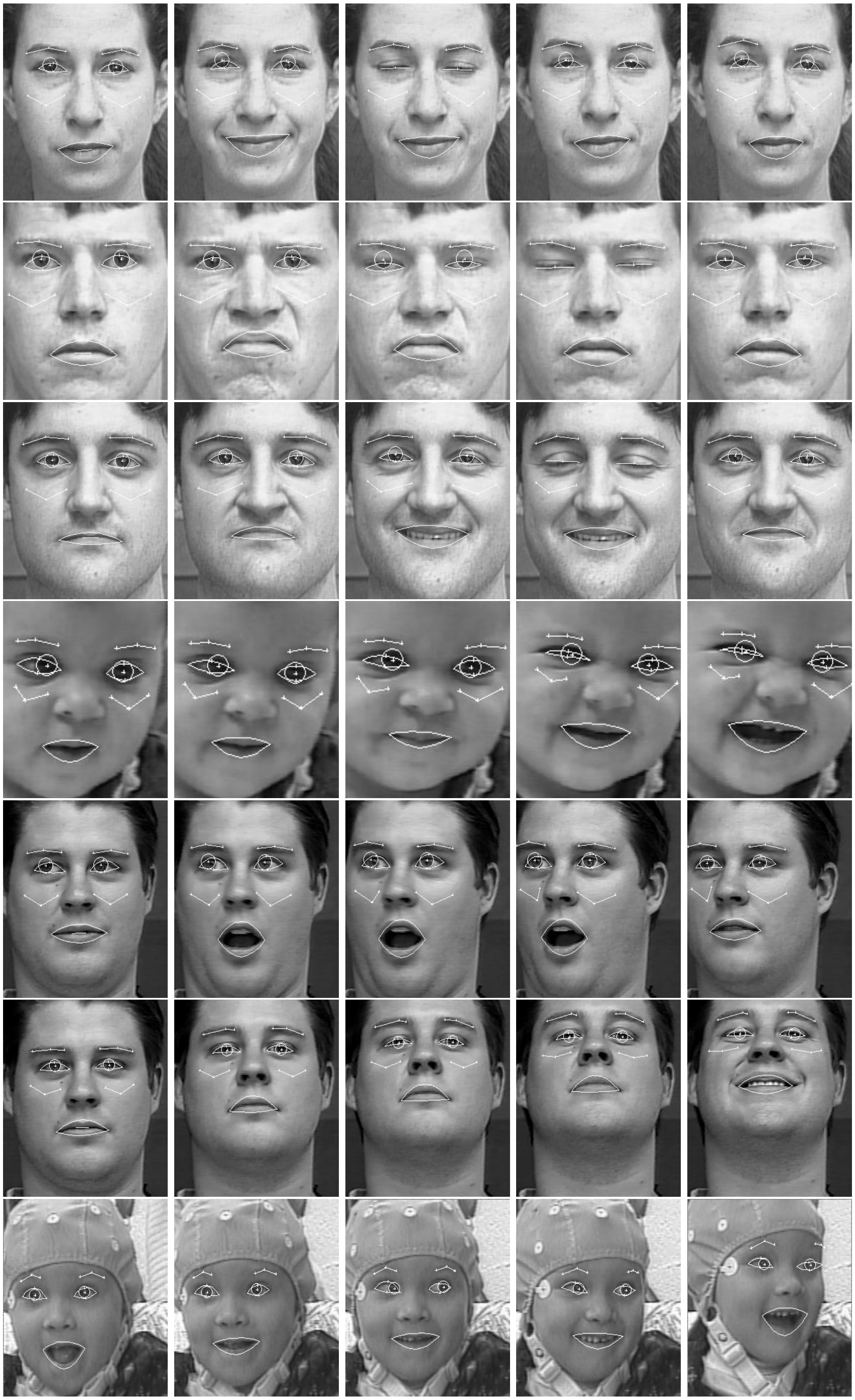
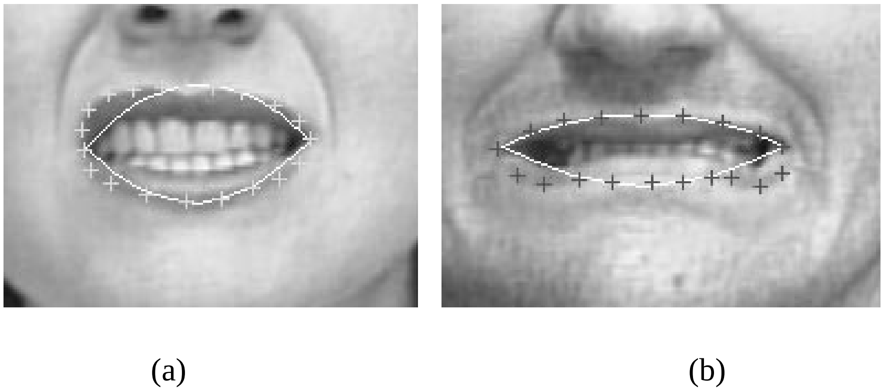


Figure 17. Tracking results by our method for different subjects. (a) Asian, (b) Afro-American, (c) and (d) Caucasian with head motion, (e) Infant with head motion.



(a) (b)

Figure 18. Non-symmetrical expression and complex Lip shapes. "+" indicate the real lip contour

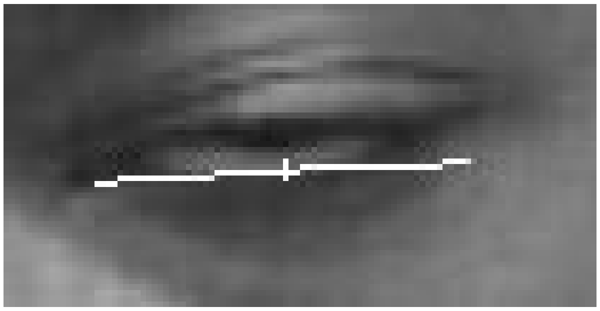


Figure 19. The narrow open eye is detected as closed because the visible iris is too small.