Dual-state Parametric Eye Tracking

Ying-li Tian [1,3] Takeo Kanade[1] and Jeffrey F. Cohn[1,2]

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213
[2] Department of Psychology, University of Pittsburgh, Pittsburgh, PA 15260
[3] National Laboratory of Pattern Recognition
Chinese Academy of Sciences, Beijing, China
Email: {yltian, tk}@cs.cmu.edu jeffcohn@pitt.edu

Abstract

Most eye trackers work well for open eyes. However, 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 tracking the locations of the person's eyes but obtain their detailed description. We need to recover the state of the eyes (i.e. whether they are open or closed), and the parameters of an eye model (e.g. the location and radius of the iris, and the corners and height of the eye opening). In this paper, we develop a dual-state model based system of tracking eye features that uses convergent tracking techniques and show how it can be used to detect whether the eyes are open or closed, and to recover the parameters of the eye model. Processing speed on a Pentium II 400MHZ PC is approximately 3 frames/second. In experimental tests on 500 image sequences from child and adult subjects with varying colors of skin and eye, accurate tracking results are obtained in 98% of image sequences.

1. Introduction

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).

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 [3, 5, 6, 8, 9, 14, 15]. Most eye feature extraction methods have been improved based on Yuille's deformable template method to track both eye locations and extract their parameters [3, 5, 9, 14, 15]. However, the deformable template scheme is time consuming and hard to be used as an eye tracker for image sequences. Also the template must be started at or below the eye, otherwise, it will locate the eyebrow instead of the eye. Chow et al. [3] 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.* [14]. Lam and Yan [9] 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 [5] *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 [10] 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 1. 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.

In the field of gesture recognition, state-based models have been used [1, 7, 13]. 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 one eye tracker which is robust to both open and closed eyes, we allow an eye to be in one of two states in each image in the

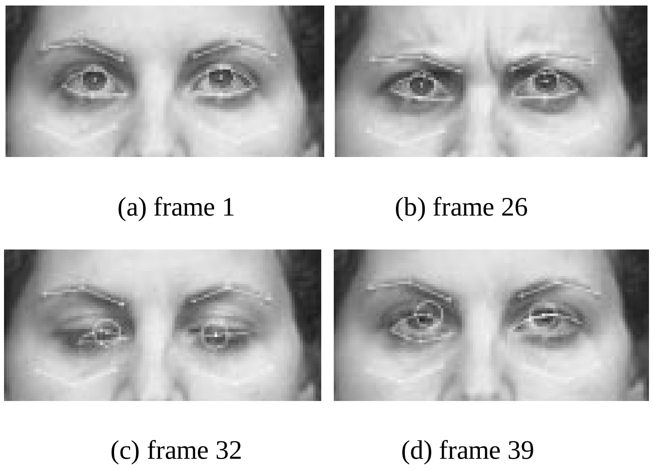


Figure 1. Eye tracking using feature point tracking [10] in the presence of eye closure. As the eyes close, the feature points disappear and so the tracker fails. After the eye opens, the iris in the left eye and the contour of the right eye have been lost.

sequence.

In this paper we propose a robust and accurate method of tracking the eye locations, detecting the eye states, and estimating the eye parameters for each frame in a sequence. Our purpose is to develop an eye tracker which is robust to blinking, and which accurately recovers the eye parameters. Several different tracking techniques are used in the system such as feature point tracking, and masked edge filtering. A dual-state eye model is used to detect the different eye states "open" or "closed". Initialized the eye template in the first frame, the eye's inner corner can be tracked accurately by feature point tracking. We assume the outer corners of eyes are in the line connecting two inner corners of eyes. Then, the outer corners can be obtained by the eye's shape information which calculated from the first frame. The edge and intensity of the iris are used to detect the eye states. For an open eye, the eyelids should be tracked by feature point tracking. For a closed eye, the eyelids contours don't need be tracked. Our eye tracking method has been tested with 500 image sequences, and it works well for 493 image sequences. The experimental results show that our method works well for both eye states, and the correct eye parameters are recovered even after the eyes close. Our tracker works robustly and accurately across identity, race, and expression. It also works well in the presence of eye making and head motion.

2. Dual-State Eye Model

2.1. Dual-State Eye Model

In the field of gesture recognition, state-based models had been used [1, 7, 13]. 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. As shown in Figure 2, we define an eye state is open or closed in the image sequences.

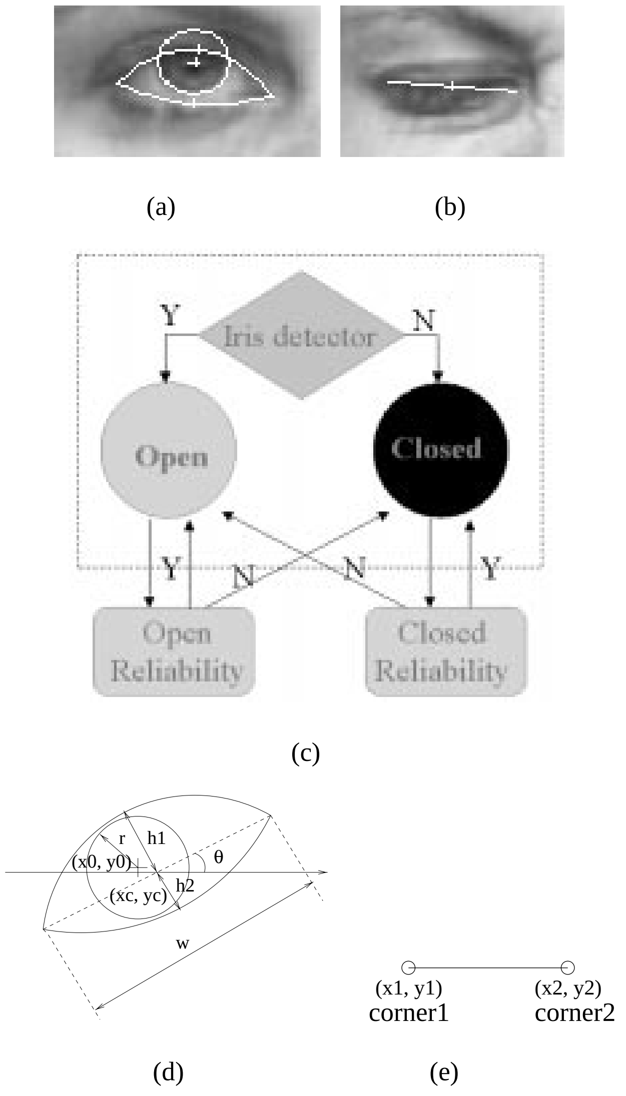


Figure 2. Dual-state eye model. (a) An open eye. (b) A closed eye. (c) The state transition diagram. (d) The open eye parameter model. (e) The closed eye parameter model.

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 specific algorithms are used to obtain eye features. These will be described in the next couple of sections.

2.2. Eye template

For an open eye, we use the same eye template as Yuille's except for two points located at the center of the whites [15]. The template, illustrated in Figure 2 (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 2 (e), is sufficient for describing closed eye features.

3. Eye state detection and eye tracking

3.1. 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 [3, 5, 9, 14, 15].

3.2. Eye region intensity normalization

For some image sequences, the eye region is very dark because of eye makeup or poor illumination. 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 image sequences, the R, G, B channels are stretched separately. In experiments, we found that our tracker works well after this intensity normalization for those images with dark eye regions.

3.3. Eye corner tracking

3.3.1. 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 [11]. 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:

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

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

$$d_{n+1} = d_n + \left\{ \sum_{x \in R} (\frac{\partial I}{\partial x})^T |_{x+d_n} [I_t(x) - I_{t+1}(x)] \right\} \left[\sum_{x \in R} (\frac{\partial I}{\partial x})(\frac{\partial I}{\partial x})^T |_{x+d_n} \right]^{-1}, \qquad (2)$$

where 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, which 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 [12]. 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. Rapid and large displacements of up to 100 pixels can be tracked robustly while maintaining sensitivity to sub-pixel facial motion.

3.3.2. 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.

3.4. Iris tracking and eye state detection

3.4.1. 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 [2] to get the eye edge maps. The edge detection results for several different open eye stages are shown in Figure 4. 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 3). 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 $r_0/3$.

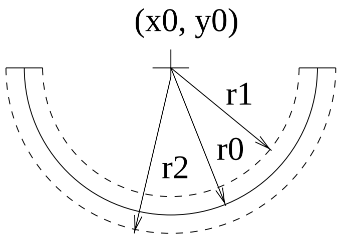


Figure 3. Half circle iris mask. (x_0, y_0) is the iris center; r_0 is the iris radius; r_1 is the minimum radius of the mask; r_2 is the maximum radius of the mask

3.4.2. 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.

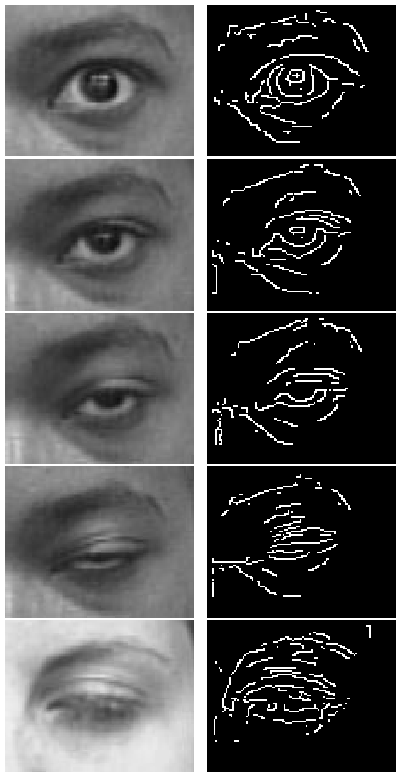
2. Extract the edge maps in the eye region and calculate the number of pixels belong to edges E_0 in the area between r_1 and r_2 of the iris mask in the first frame.

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 (3).

$$Eyestate = \begin{cases} Open & \text{if the iris detected} \\ Closed & \text{otherwise} \end{cases} \quad (3)$$



(a) Original eye images. (b) Eye edge maps.

Figure 4. 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.

3.5. Eye boundary tracking

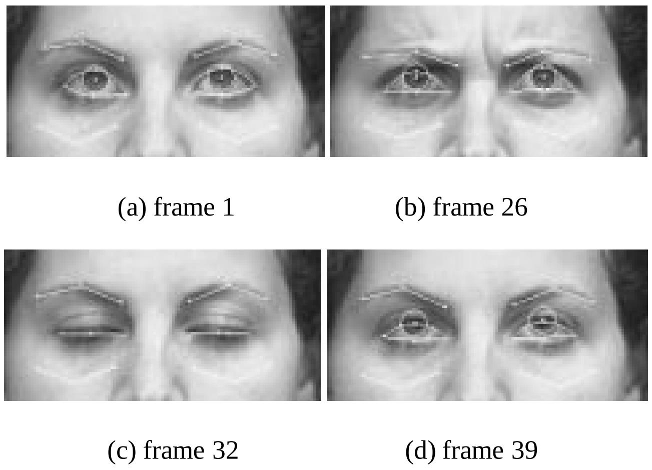
For open eye boundary tracking, the eye template is used to obtain the correct eye boundaries. After locating the eye template in the first frame, only two key points of the eye template are tracked (the center points of the upper and the lower eyelids) in the remaining open eye images. From these two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

4. Experiment results

The dual-state eye tracking method was tested using 500 image sequences from the Pitt-CMU Facial Expression AU

Coded Database [4]. 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 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. Generally, the size of a full-face is about 220x300 pixels and the size of one eye region is about 60x30 pixels.


(a) frame 1 (b) frame 26

(c) frame 32 (d) frame 39

Figure 5. 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.

Compared to Figure 1, the eye tracking results for images in the sequence after the eye is closed are given in Figure 5. The iris and eye contours are tracked correctly by our method. Figure 6 (a) shows eye tracking result for narrow open eyes. The wide open eye and blink tracking results are demonstrated in Figure 6 (b). Figure 7 shows the eye tracking results for tightly closed eye and blink. The correct iris position and eye boundaries were tracked after the eye was closed multiple times. Notice the semi-circular iris model accurately tracks the iris even when it is only partially visible. Figure 8 demonstrates the robustness of our algorithm in the performance of iris, head, and background motion. More tracking results are available on http://www.cs.cmu.edu/∼face/. Processing speed on a Pentium II 400MHZ PC is approximately 3 frames/second. Our method works well for men, women, and children, and across skin color and appearance. We tested our method in 500 image sequences. Of these, good results were obtained in 493 sequences.

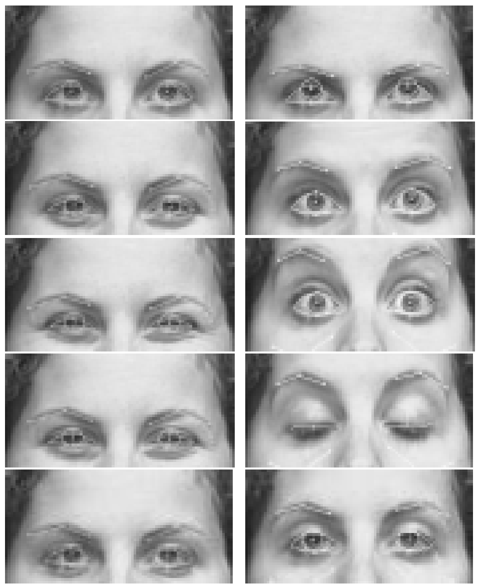

(a) narrow eye (b) wide open eye

Figure 6. Tracking results by OUR METHOD for narrow open eye and wide open eye include blinking.

5. Conclusion and discussion

We have described a dual-state eye tracking method for recovering the parameters and state of an eye model as well as tracking the locations of the eyes in an image sequences for various subjects with various expressions. Two eye states were considered in the dual-state eye model: open and closed. Given the initial location of the eye template in the first frame, the iris was detected and tracked by intensity and edge information and then the eye state was obtained by the iris detection. If the iris is detected, the eye is open. Next, the eyelid center points were tracked for open eyes via the tracking method developed by Lucas-Kanade[11] in the image sequence and the eye contours were obtained by calculating the correspondence eye template parameters.

The experimental result show that our method works well for the long image sequence regardless of limited head rotation. However, for a very narrow open eye, for example, the fourth row of Figure 4, 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, two eye states can not provide a practical match. More eye states are needed to deal with these difficult cases. Future work will need to obtain a face state and detect the reliability of eye states.

Acknowledgements

The authors would like to thank 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] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis Mach. Intell.*, 8(6), 1986.

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

[4] 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.

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

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

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

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

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

[10] 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.

[11] 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.

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

[13] 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.

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

[15] 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.

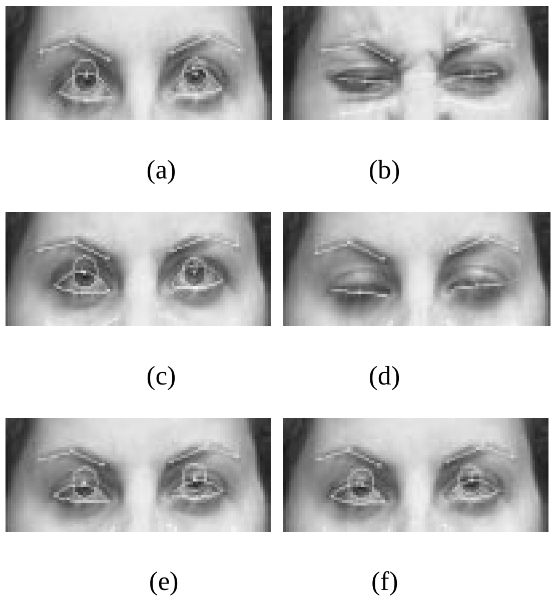


Figure 7. Tracking results by OUR METHOD for tightly closed eye and multiple blinks in a long image sequence. The eye states are correctly detected and eye parameters are accurately tracked after the eyes are tightly closed and blink. Notice how the semi-circular iris model accurately tracks the iris even when it is only partially visible.

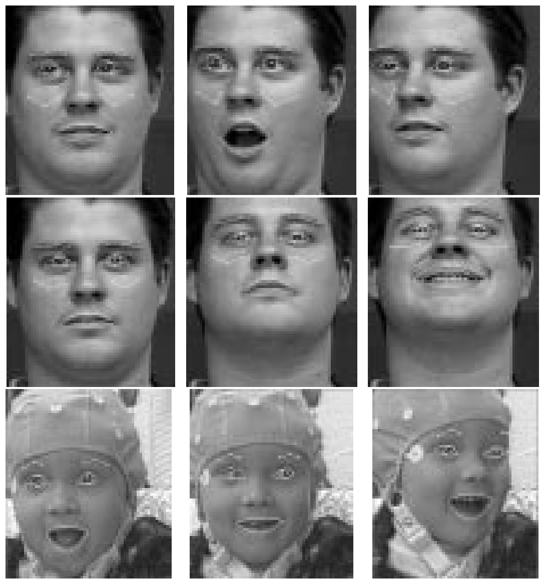


Figure 8. Eye tracking results by OUR METHOD for subjects with varying head motion and orientation.