Tracking Iris Contour with a 3D Eye-Model for Gaze Estimation

Haiyuan Wu, Yosuke Kitagawa, Toshikazu Wada,
Takekazu Kato, and Qian Chen

Faculty of Systems Engineering, Wakayama University, Japan

Abstract. This paper describes a sophisticated method to track iris contour and to estimate eye gaze for blinking eyes with a monocular camera. A 3D eye-model that consists of eyeballs, iris contours and eyelids is designed that describes the geometrical properties and the movements of eyes. Both the iris contours and the eyelid contours are tracked by using this eye-model and a particle filter. This algorithm is able to detect "pure" iris contours because it can distinguish iris contours from eyelids contours. The eye gaze is described by the movement parameters of the 3D eye model, which are estimated by the particle filter during tracking. Other distinctive features of this algorithm are: 1) it does not require any special light sources (e.g. an infrared illuminator) and 2) it can operate at video rate. Through extensive experiments on real video sequences we confirmed the robustness and the effectiveness of our method.

1 Introduction

The goal of this work is to realize video rate tracking of iris contours and eye gaze estimation with a monocular camera in a normal indoor environment without any special lighting. Since blinking is a physiological necessity for human, it is also a goal of this work to be able to cope with blinking eyes.

To detect or to track eyes robustly, many popular systems use infrared light (IR) [1][2] and stereo cameras [3][4]. Recently, mean-shift filtering [5], particle filtering [6] and K-means clustering [15] are also used for tracking eyes. The majority of the proposed methods assume open eyes, and most of them neglect the eyelids and assume that iris contours show circles in the image. Therefore, they can not detect pure iris contours which are important for eye gaze estimation.

The information about eyes used for eye gaze estimation can roughly be divided into three categories: (a) global information, such as active appearance model (AAM) [5][7], (b) local information, such as eye corners, iris, and mouth corners [8][9][10], and (c) the shape of ellipses fitted to the iris contours [11]. There are also some methods based on the combinations of them [12].

D.Hansen et al. [6] employ an active-contour method to track iris. It is based on a combination of a particle filter and the EM algorithm. The method is robust against the changes of lighting condition and camera defocusing. A looking calibration is necessary for eye gaze estimation.

Y. Yagi et al. (Eds.): ACCV 2007, Part I, LNCS 4843, pp. 688–697, 2007.

Y.Tian et al. [13] propose a method of tracking the eye locations, detecting the eye states, and estimating the eye parameters. They develop a dual-state eye model and use it to detect whether an eye is *open* or *closed*. However, it can not estimate eye gaze correctly.

It is difficult to track iris contours consistently and reliably because they are often partly occluded by the upper and the lower eyelids and confused with the eyelid contours. Most conventional methods lack the ability to distinguish iris contours from eyelid contours. Some methods separate the iris contours from eyelid contours heuristically.

In order to solve this problem, we design a 3D eye-model that consists of eyeballs, irises and eyelids. We use this model together with a particle filter to track both the iris contours and the eyelid contours. With this approach, the iris contours can be distinguished from the eyelid contours and both of them can be tracked simultaneously. Then the shape of the iris contours can be estimated correctly by only using the edge points between upper and lower eyelids, which are then used to estimate the eye gaze.

In this method, we assume that movements of the two eyes of a person are synchronized and use this constraint to restrict the movement of eyeballs in the 3D eye-model during tracking. This increases the robustness of tracking and the reliableness of eye gaze estimation of our method. Implementing on a PC with a Pentium4 3GHz CPU, the processing speed is about 30frames/second.

2 3D Eye-Model for Tracking Iris Contours

Our 3D eye-model consists of two eyes, each eye consists of an eyeball, an iris, an upper and a lower eyelids. As shown in figure 1(a), the two eyeballs are assumed as spheres of equal radius r_w. The irises are defined as circles of equal radius r_b on the surface of each eyeball. Then the distance between the eyeball center and the iris plane is

$$z_{pi} = \sqrt{r_w^2 - r_b^2}. \tag{1}$$

The centers of the two eyeballs ($\mathbf{c}_l$ and $\mathbf{c}_r$) are put on the X-axis of the eye-model coordinates system symmetrically about the origin, and the distance between each center and the origin is w_x.

$$\mathbf{c}_l = \begin{pmatrix} -w_x \; 0 \; 0 \end{pmatrix}^T; \qquad \mathbf{c}_r = \begin{pmatrix} w_x \; 0 \; 0 \end{pmatrix}^T. \tag{2}$$

We define direction of the visual lines when people look forward same as the Z-axis. Then the plane where the iris resides will be parallel to the X-Y plane. The iris contours of the left and the right eye ($\mathbf{p}_l$ and $\mathbf{p}_r$) can be expressed with the following expressions.

$$\mathbf{p}_j(\alpha) = \mathbf{p}(\alpha) + \mathbf{c}_j, \quad j = l, r \tag{3}$$

where

$$\mathbf{p}(\alpha) = \begin{pmatrix} r_b \cos\alpha & r_b \sin\alpha & z_{pi} \end{pmatrix}^T, \quad \alpha \in [0, 2\pi].$$

The upper and lower eyelids are defined as B-spline curves located on the plane $z = r_w$, which is a vertical plane in front of eyeballs. As shown in figure 1(a), each eyelid has three control points. We let the upper and the lower eyelid of the same eye share the same inner eye corners ($\mathbf{E}_{hl}$ and $\mathbf{E}_{hr}$) and the outside eye corners ($\mathbf{E}_{tl}$ and $\mathbf{E}_{tr}$). The eight control points ($\mathbf{E}_{hl}$, $\mathbf{E}_{ul}$, $\mathbf{E}_{tl}$, $\mathbf{E}_{dl}$, $\mathbf{E}_{hr}$, $\mathbf{E}_{ur}$, $\mathbf{E}_{tr}$ and $\mathbf{E}_{dr}$) describe the shape of the eyelids when the two eyes are open. Since the values of these eight control points, w_x, r_w and r_b depend on each individual person, we call them as personal parameters which are estimated at the beginning of tracking.

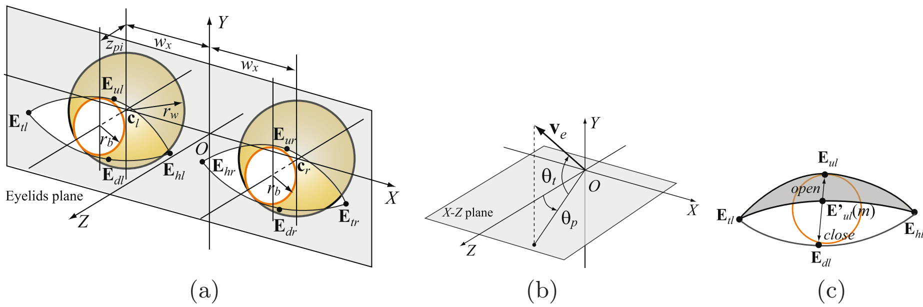


Fig. 1. (a) The structure of the 3D eye-model. (b) The gaze vector and the movement parameters of an eyeball. (c) The movement parameter (m) of an upper eyelid.

In general, when people look at a far place, the visual lines of both eye will be approximately parallel. Also, when people blink, in most cases the lower eyelids do not move and the upper eyelids of the two eyes move in the same way. In this paper, in order to keep the number of eye movement parameters small so that the particle filter can work efficiently while not losing generrality, we assume that the lower eyelids keep still and the movements of the eyeballs and the upper eyelids of two eyes are same and synchronized. Therefore, only the parameters for describing the movements of the eyeball and the two eyelids of one eye are necessary because the movement of the another eye can be described with the same parameters.

The movement of the left (or right) eyeball can be described by an unit vector $\mathbf{v}_e$ that indicates the gaze direction (See figure 1(b)).

$$\mathbf{v}_e = \begin{pmatrix} \cos\theta_t \sin\theta_p & \sin\theta_t & \cos\theta_t \cos\theta_p \end{pmatrix}^T , \tag{4}$$

where θ_t is the angle between $\mathbf{v}_e$ and the X-Z plane (tilt), and θ_p is the angle between the projection of $\mathbf{v}_e$ onto the X-Z plane and Z-axis (pan). In this paper, θ_t and θ_p are used as the movement parameters of the left and the right eyeball. Then the iris contours of the left and the right eye can be expressed by

$$\mathbf{p}_j^M(\alpha, \theta_t, \theta_p) = \mathbf{R}(\theta_t, \theta_p)\mathbf{p}(\alpha) + \mathbf{c}_j; \quad j = l, r, \tag{5}$$

where

$$\mathbf{R}(\theta_t, \theta_p) = \begin{pmatrix} \cos\theta_p & 0 & \sin\theta_p \\ -\sin\theta_t \sin\theta_p & \cos\theta_t & \sin\theta_t \cos\theta_p \\ \cos\theta_t \sin\theta_p & -\sin\theta_t & \cos\theta_t \cos\theta_p \end{pmatrix}. \tag{6}$$

The movement of opening or closing eyes can be expressed by changing the shape of the upper eyelids. This movement is expressed by movement of the middle control point of an upper eyelid ($\mathbf{E}'_{ul}$ and $\mathbf{E}'_{ur}$) along the line connecting the middle control points of the upper and the lower eyelids of an opened eye as shown in figure 1(c). The middle control points of the upper eyelids can be expressed as following

$$\mathbf{E}'_{uj} = m\mathbf{E}_{uj} + (1-m)\mathbf{E}_{dj}; \quad j = l, r, m = 0 \sim 1, \tag{7}$$

where m is a parameter that describes the movements of the both upper eyelids.

The parameters θ_t, θ_p and m are movement parameters of our 3D eye-model. In order to track the iris contours and estimate the eye gaze with a particle filter, it is necessary to project the iris contours and the eyelid contours onto the image plane. This can be done with the following equation.

$$\mathbf{i}_p = \mathbf{M}_c(\mathbf{R}_h \mathbf{p} + \mathbf{T}_h). \tag{8}$$

$\mathbf{p}$ is the point on an iris contour or an eyelid contour in the 3D eye-model and $\mathbf{i}_p$ is its projection onto the image plane. $\mathbf{M}_c$ is the projection matrix of the camera which is assumed known. $\mathbf{R}_h$ and $\mathbf{T}_h$ are the rotation matrix and the translation vector of the eye-model coordinates system relateive to the camera, which are the movement parameter of the head and also are the ones to be estimated with the particle filter during tracking.

3 Likelihood Function for Tracking Iris

In many applications using the particle filter (also called as Condensation[14]), the distance between a model and the edges detected from an input image is used as the likelihood. However, in the case of iris contour tracking, this definition of likehood often leads particles to converge on a wrong place (such as inner eyes or eyelashes). This is because there are many edges so that the likelihood also becomes high at those places.

In order to track the iris contours with the particle filter, we define a likelihood function by considering both the image intensity and the image gradient.

3.1 The Likelihood Function of Irises

In most cases, the brightness of iris in an input image is lower than its surroundings. In order to make use of this fact, the average brightness of iris area is introduced into the likelihood function of iris. In this paper, we let the the average brightness the iris area of 3D eye model be 0. Then the values (E_l and

E_r) indicating the likehood of an iris candidate area in an image by considering the image intensity can be calculated with the following equation.

$$E_j = e^{-Y_j^2/k}, \qquad j = l, r. \tag{9}$$

Here, Y_l, Y_r are averages of the brightness of the left and right iris candidate areas, and k is a constant. The higher the average brightness in those areas are, the lower E_l and E_r will be.

In order to reduce the influence of the non-iris contour edges when estimating the likelihood for irises, we consider the direction of the edges as well as their strength. Since an iris area is darker than its surrounding, the direction of the image gradient at iris contour will be outward from the iris center. In this paper, we define the direction of the normal vector of the iris contour in our 3D eye-model is outward from the iris center. Therefore, if the iris contour of the 3D eye-model and the iris contour in the image overlaps, the direction of the normal vector and the image gradient will be same.

We pick n points from the iris contours of the 3D eye-model at fixed intervals as following.

$$\mathbf{p}_{jk}^s = \mathbf{p}_j^M \left(\frac{2\pi k}{n}\right), \quad j = l, r, \quad k = 0, 1, \cdots, n-1. \tag{10}$$

These points are called *iris contour points* as (*ICP*s). By using the hypothesis generated by the particle filter, which is a set of parameters describe the movements of the eyes and the head, the projection of ICPs ($\mathbf{i}_{jk}^s$) and the normal vectors ($\mathbf{h}_{jk}$) of them on the image plance can be calculated. Let $\mathbf{g}(\mathbf{i}_{jk}^s)$ indicates the image gradient at each projected ICP, the likelihood π_I of iris candidates in the image is computed with the following expression.

$$\pi_I = E_l \frac{\sum_{k=0}^{n-1} B(\mathbf{i}_{lk}^s) D(\mathbf{i}_{lk}^s)}{\sum_{k=0}^{n-1} B(\mathbf{i}_{lk}^s)} + E_r \frac{\sum_{k=0}^{n-1} B(\mathbf{i}_{rk}^s) D(\mathbf{i}_{rk}^s)}{\sum_{k=0}^{n-1} B(\mathbf{i}_{rk}^s)}, \tag{11}$$

here, $B(\mathbf{i}_{jk}^s)$ is a function for removing the influence of the edges of eyelids by ignoring the ICPs outside the region enclosed by the lower and the upper eyelids,

$$B(\mathbf{i}_{jk}^s) = \begin{cases} 1 & \text{if } \mathbf{i}_{jk}^s \text{ is between the lower and the upper eyelids} \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

and

$$D(\mathbf{i}_{jk}^s) = \begin{cases} \mathbf{h}_{jk} \cdot \mathbf{g}(\mathbf{i}_{jk}^s) & \text{if } \mathbf{h}_{jk} \cdot \mathbf{g}(\mathbf{i}_{jk}^s) > 0 \\ 0 & \text{otherwise} \end{cases}, \; j = l, r. \tag{13}$$

Here, $\cdot$ indicates inner product.

3.2 The Likelihood Function of Eyelids

In order to calculate the likelihood of iris more correctly, it is also necessary to track the eyelids. Since the difference between the brightness of the two sides of an eyelid is not as big as the case of an iris contour, we only use the image gradient to estimate the likelihood of eyelids (π_E).

$$\pi_E = \frac{\sum_{k=0}^{N-1} D(\mathbf{i}_{lk}^d)}{N} + \frac{\sum_{k=0}^{N-1} D(\mathbf{i}_{rk}^d)}{N}, \tag{14}$$

here

$$D(\mathbf{i}_{jk}^d) = \begin{cases} \mathbf{h}(\mathbf{i}_{jk}^d) \cdot \mathbf{g}(\mathbf{i}_{jk}^d) & \text{if } \mathbf{h}(\mathbf{i}_{jk}^d) \cdot \mathbf{g}(\mathbf{i}_{jk}^d) > 0 \\ 0 & \text{otherwise} \end{cases}, \quad j = l, r,$$

$\mathbf{i}_{jk}^d$ is the projection of each point on the eyelids and $\mathbf{h}_{jk}^d$ is its normal vector. The likelihood function π of the whole 3D eye-model including the irises and the eyelids is defined with the following expression.

$$\pi = \pi_I \pi_E \tag{15}$$

4 Eye Gaze Estimation

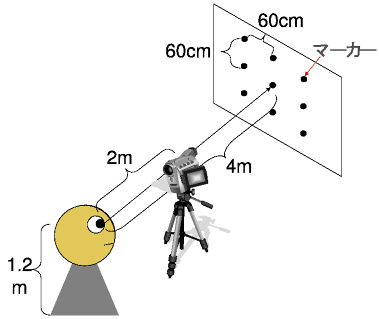
In order to estimate the eye gazes with the 3D eye-model, it is necessary to determine the personal parameters described in section 2 for testee's eyes. At the begining of tracking, we assign the positions of the inner eye corners ($\mathbf{E}_{hl}$ and $\mathbf{E}_{hr}$), and the outside eye corners ($\mathbf{E}_{tl}$ and $\mathbf{E}_{tr}$) on the image manually. The other parameters of eyelids are drawn from these four points. The personal parameters about eyeballs are set to the average values of people. After this, all personal parameters except the four manually assigned points are estimated with a particle filter by using the several frames of image. During tracking, the movement parameters of the eyes and the head that give the maximum value of likelihood are estimated by the particle filter. From these movement parameters the eye gaze is calculated.

5 Experimental Results

We have tested our method of tracking iris contours and eye gaze estimation using a PC with a 3GHz Pentium 4 CPU. The input image (640×480 pixels) sequences were taken by a normal video camera. The experimental environment is show in figure 2(a).

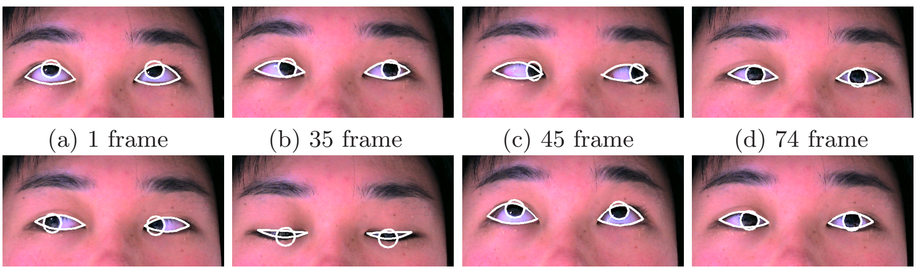
5.1 Tracking the Iris Contour

Firstly, we used our method to track iris contours. The parameters of particle filter are θ_p, θ_t, m, $\mathbf{T} = (t_x, t_y, t_z)$, and ψ that is the rotation angle of the head (eye-model) around z axis of the camera coorinates system. As for the random sampling, the standard deviations of normal distribution were taken as θ_p: 5[degree], θ_t: 5[degree], m: 0.05, t_x: 0.1[cm], t_y: 0.03[cm], t_z: 0.01[cm], and ψ: 0.001[rad], respectively.



(a) (b)

Fig. 2. (a): The experimental environment. (b): The environment for evaluating the accuracy of eye gaze.



(a) 1 frame (b) 35 frame (c) 45 frame (d) 74 frame

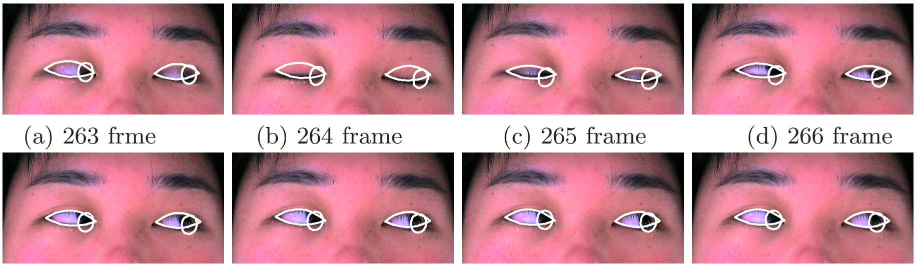
(e) 80 frame (f) 140 frame (g) 170 frame (h) 200 frame

Fig. 3. Some tracking results with random sampling twice for each frame by 150 samples

The processing time of the tracking was 9.6ms/frame for 100 samples, and was 30.8ms/frame for 400 samples, respectively. When the eye moved quickly, a lot of delays occured during tracking in the case of 100 samples, and only a little occured in the case of 400 samples.

In order to increase the tracking accuracy, we carried out the random sampling twice for each frame by 150 samples. Some tracking results are shown in figure 3. In this case, the processing time was 29ms/frame and the iris contour and the eyelid contour could be tracked without delay even when the eyes moved quickly. Moreover, the tracking accuracy was much improved. All experiments described hereafter were carried out in this way.

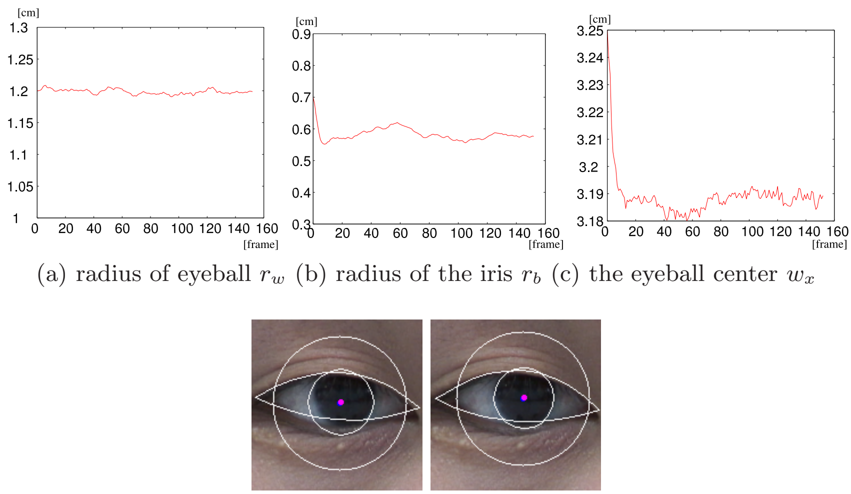
When a person was blinking the eyes, as shown in figure 4, the eyelids moved quickly thus could not be tracked perfectly. Also, since the irises were not visible when the eyes were closed, the iris contours could not be tracked exactly. In the case that the system has detected closed eyes (from the movement parameter m), the system holds the former state of the irises just before the eyes were closed. When the irises become visible after blinking, the tracking for irises will



(a) 263 frme (b) 264 frame (c) 265 frame (d) 266 frame

(e) 267 frame (f) 268 frame (g) 269 frame (h) 270 frame

Fig. 4. Some tracking results of blinking eyes



(a) radius of eyeball r_w (b) radius of the iris r_b (c) the eyeball center w_x

(d) first frame (e) 15 frame

Fig. 5. The convergence of personal parameters

be restarted again. From this experimental resuls we have confirmed that our method can track iris contours even when people blink their eyes.

5.2 Initialization of Personal Parameters

After the four eye corners had been given manually, the rest personal parameters of the 3D eye-model were estimated with a particle filter. As for the random sampling, the standard deviations of normal distribution were taken as r_b: 0.02[cm], r_w: 0.01[cm], w_x: 0.03[cm], all the x and y coordinats of the control points of eyelids: 0.2[cm] and 0.05[cm], respectively. The random sampling by 1500 samples was performed to each frame, and the likelihood was evaluated twice. An known eye gaze is necesary for estimating the personal parameters. In the experiments, this estimation was carried out using the images when the person looked forward.

Figure 5 shows the behavior of estimated values of some personal parameters. The horizontal axis indicates the frame number and the vertical axis indicates the estimated value. From figure 5, we confirmed that the personal parameters converge within several frames. Figure 5(d) and (e) show the estimated results of the initial frame and the 15th frame.

The processing time for estimating the personal parameters was 430ms/frame. After that, our algorithm could work at video rate (30.1ms/frame) for tracking iris contour and estimating eye gaze.

5.3 Accuracy Evaluation of Eye Gaze Estimation

In order to evaluate the accuracy of the estimated eye gaze using the proposal method, we put some markers on a wall which was 4 meters away from a testee(see figure 2(b)). The number of testee was five. We leted the testee gaze at each marker for 2 second and estimated the eye gaze with our system. Table 1 shows the difference of the mean value of estimated eye gaze and the true value of each marker.

Table 1. The difference of the mean value of estimated eye gaze and the true value of each marker to (x-direction, y-directions) [unit: degree]

Person A	1 low	2 low	3 low
Upper col.	(0.6,-0.5)	(1.5,1.9)	(2.5,1.5)
Middle col.	(2.6,2.4)	(2.5,1.6)	(1.8,0.6)
Bottom col.	(4.9,5.1)	(1.9,3.6)	(1.7,4.3)

Person B	1 low	2 low	3 low
Upper col.	(-1.8,-0.5)	(-2.9,-0.1)	(-4.1,1.4)
Middle col.	(-1.6,1.7)	(-1.6,2.1)	(-4.8,1.6)
Bottom col.	(-0.9,5.4)	(-2.4,6.0)	(-3.4,6.0)

Person C	1 low	2 low	3 low
Upper col.	(5.1,3.7)	(2.5,3.3)	(2.6,1.3)
Middle col.	(2.4,-1.8)	(2.9,-1.4)	(0.9,-0.2)
Bottom col.	(1.9,1.2)	(-2.0,1.9)	(-3.2,2.9)

Person D	1 low	2 low	3 low
Upper col.	(-3.3,-1.8)	(1.1,0.0)	(-1.0,0.0)
Middle col.	(-4.5,-1.8)	(-2.4,-1.3)	(1.6,-0.1)
Bottom col.	(0.9,0.4)	(-0.9,0.2)	(0.1,0.8)

Person E	1 low	2 low	3 low
Upper col.	(3.5,0.8)	(2.4,1.4)	(4.5,1.5)
Middle col.	(-0.9,-5.3)	(1.2,-6.4)	(3.5,-3.5)
Bottom col.	(3.8,-1.6)	(-2.6,-1.9)	(-1.1,-2.2)

6 Conclusion

This paper aims at tracking iris contours and estimating eye gaze from monocular video images. In order to suppress the influence of eyelid edges, we have proposed a 3D eye-model for tracking iris and eyelid contours. Using this eye-model, the eyelid contours and iris contours can be distinguished. By only using the edge points between upper and lower eyelids, the shape of the iris can be estimated and then the eye gaze can be measured. From the experimental results, we confirmed that the proposed algorithm can track iris contours and eyelid contours robustly and can estimate the eye gaze at video rate. The proposal algorithm can be used to various applications, such as, to check on whether looking-aside movement of a driver, etc.

Acknowledgments. This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 18200131 and 19500150.

References

1. Zhu, Z., Ji, Q.: Eye Gaze Tracking Under Natural Head Movements. In: CVPR, vol. 1, pp. 918–923 (2005)
2. Hennessey, C., Noureddin, B., Lawrence, P.: A Single Camera Eye-Gaze Tracking system with Free Head Motion. In: Symposium on Eye tracking research & applications, pp. 87–94 (2006)
3. Matsumoto, Y., Zelinsky, A.: An Algorithm for Real-time Stereo Vision Implementation of Head Pose and Gaze Direction Measurement. In: FG, pp. 499–504 (2000)
4. Beymer, D., Flickner, M.: Eye Gaze Tracking Using an Active Stereo Head. In: CVPR, vol. 2, pp. 451–458 (2003)
5. Hansen, D., et al.: Tracking Eyes using Shape and Appearance. In: IAPR Workshop on Machine Vision Applications, pp. 201–204 (2002)
6. Hansen, D.W., Pece, A.: Eye Typing off the Shelf. In: CVPR, vol. 2, pp. 159–164 (2004)
7. Ishikawa, T., et al.: Passive Driver Gaze Tracking with Active Appearance. In: 11th Word Congress on ITS in Nagoya, pp. 100–109 (2004)
8. Gee, A., Cipolla, R.: Estimating Gaze from a Single View of a Face. In: ICPR, pp. 758–760 (1994)
9. Zhu, J., Yang, J.: Subpixel Eye Gaze Tracking. In: FG (2002)
10. Smith, P., Shah, M., Lobo, N.: Determining Driver Visual Attention with One Camera. IEEE Trans. On Intelligent Transportation System 4(4), 205–218 (2003)
11. Wu, H., Chen, Q., Wada, T.: Visual Direction Estimation from a Monocular Image. IEICE E88-D(10), 2277–2285 (2005)
12. Wang, J.G., Sung, E., Venkateswarlu, R.: Eye Gaze Estimation from a Single Image of One Eye. In: ICCV (2003)
13. Tian, Y.l., Kanade, K., Cohn, J.F.: Dual-state Parametric Eye Tracking. In: FG, pp. 110–115 (2000)
14. Isard, M., Blake, A.: Condensation-conditional density propagation for visual tracking. IJCV 29(1), 5–28 (1998)
15. Hua, C., Wu, H., Chen, Q., Wada, T.: A General Framework For Tracking People. In: FG, pp. 511–516 (2006)
16. Duchowski, A.T.: A Breadth-First Survey of Eye Tracking Applications, Behavior Research Methods, Instruments, and Computers (2002)
17. Criminisi, A., Shotton, J., Blake, A., Torr, P.H.S.: Gaze Manipulation for One-to-one Teleconferencing. In: ICCV (2003)
18. Yoo, D.H., et al.: Non-contact Eye Gaze Tracking System by Mapping of Corneal Reflections. In: FG (2002)
19. Schubert, A.: Detection and Tracking of Facial Feature in Real time Using a Synergistic Approach of Spatial-Temporal Models and Generalized Hough-Transform Techniques. In: FG, pp. 116–121 (2000)