Proceedings of the 2013 IEEE/SICE International Symposium on System Integration, Kobe International Conference Center, Kobe, Japan, December 15-17,

MA1-K.1

Eyelid and Iris Tracking Method with Novel Eye Models

Kimimasa Tamura and Yoshimitsu Aoki

Abstract— **These days, useful gaze estimation system is getting expected as a new interface. However previous most gaze estimation systems require special equipment or complex calibration. In this report, we propose the method which tracks user's eyelid and iris automatically with novel eyelid model and eyeball model, aiming at useful gaze estimation as a future work. Eyelid shape tracking is also useful for more accurate iris tracking and user's state estimation.**

I. Introduction

WITH the diffusion of PC and cameras, gaze estimation technique is getting more popular as a new tool for human computer interaction. Gaze information includes user's idea and interest, so it is useful for marketing analysis tool in store, effect measurement for digital signage. In driving situation, this technique is helpful to prevent traffic accident by monitoring driver's gaze direction and awareness and alert inattentive driving. Gaze estimation is also expected as a new interface on various devices. Most commercial PC and mobile devices have built-in camera, so new interface with gaze estimation can be achieved by adding only software. This is appreciated by persons with handicap.

Considering these situations in daily lives, it is needed to estimate unspecified person's gaze direction only with an ordinary camera. It also should deal with head pose movement and illuminated changes.

A. Related Work

Conventional gaze estimation methods have two main streams. The first one uses infrared LED and the other uses invasive device like head mounted device.

The first method spotlight on user's cornea with infrared LED, measuring its reflection point and iris position then it estimates user's gaze direction. This method has very high accuracy and suitable with laboratory environment. It needs, however, special equipment like infrared LED and camera. In addition to this, it limits the place, where the light can reach and not be vanished by sun light.

The second method witch is invasive has a merit that the sight from camera is the same one from eye. But it is impossible to force users to wear head mounted devices in daily scene. Of course it's become problem that these devices prevent users' natural gaze movement.

To achieve the convenient gaze estimation system in daily lives, the system can use only one ordinary camera and get rid of user's loads as many as possible. There are two approaches suitable with these conditions. First is appearance based approach, which learns the relation between eye image and gaze point. Second is called model based approach, which uses genetic eye ball model to estimate gaze direction.

In the appearance based methods, Sugano et al.[4][5]proposed an automatic machine learning method. It considers user's clicking point on display as the gazed point, using the relation as a supervised sample. Continuing clicking, the accuracy of gaze estimation gets high. They have another approach of automatic learning: they assume that users gaze at visually salient point in the scene. Thanks to these approach, the achieved gaze estimation only with one ordinary camera without any user's special action for calibration. This approach needs hundreds of samples, however, to get enough accuracy. It is very sensitive against user's head movement because it directory uses eye image as feature information.

The model based approach, comparing bio models including face and eye and actual input image, estimates gaze direction. The good points of this method is that it uses the eye ball feature which has not much variation among persons, it can get enough accuracy without personal calibration. If head pose changes, this model easily handle this head pose rotation. But there are some challenges how it can track feature points in noisy and less resolution image. These feature points' accuracy, including iris and eye corners, are very important in gaze estimation. Yamazoe[6] proposed the method, which compares projected image from 3D eye ball model with white and black parts and input image. Then it estimates gaze direction by estimating suitable eye ball parameters minimizing errors. Kitagawa[7]'s method uses eye ball model with simple eyelid shape. This method achieved gaze estimation with monocular camera permitting small head pose changes. However, eyelid shapes are complex and vary among persons, it is not enough to approximate eyelid line in parabola line. It permits only frontal face, not automatic system because we have to click four points which represent eye position and shape. In Zhang's method[8] it estimates gaze points from iris's ellipse parameters by ellipse fitting. This method uses only ellipse shape information, so it needs very high resolution eye image to get enough accuracy.

Considering the above, we propose a method that tracks user's eyelid and iris automatically and instantly from an ordinary camera. This is the pre-processing for the gaze estimation easy to use. Our method takes model base approach because we have to deal with head pose changes and achieve automatic and instant tracking.

This paper is written as below; in chapter 2 we show brief outline of this method, in chapter 3, 4, 5 we describe each method of head pose, eyelid, iris tracking in detail. Chapter 6

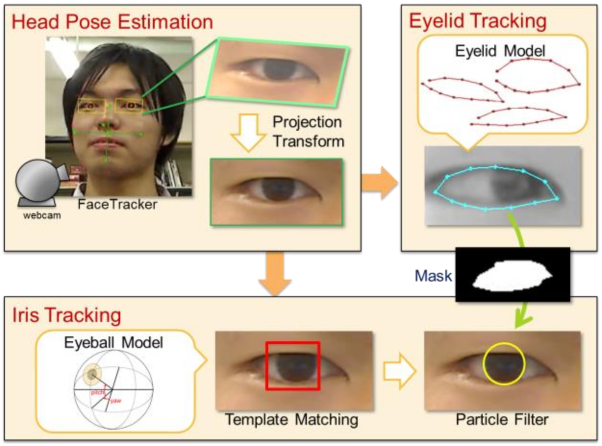


Fig. 1. System Flow

is about experiment and we discuss in chapter 7.

II. Proposal Method

Our system flow is shown in Fig. 1. First of all, we apply FaceTracker[9] to the input face image in order to detect a face and to extract facial feature points. Then from the sets of 2D facial feature points, we reconstruct individual user's 3D face model, using which we estimate user's head pose(section B). We crop an eye region image from the input face image based on the head pose, and perform Eyelid tracking in the eye image. Particle Filter is used in the Eyelid Tracking process. On ahead, we made the Eyelid Shape Model that represents eyelid shape changes. Based on this model, many particles are spread on an eye image and Eyelid Tracking is performed. Then, we perform Iris Tracking in the same eye image using template matching and Particle Filter.

III. Head Pose Estimation

In order to achieve head pose invariant gaze estimation, head pose information is necessary. Head pose is also necessary for Eyelid Tracking described in next section. This is because the appearance of eyelid changes when head pose changes even if the state of eyelid doesn't change, so we have to compensate the appearance changes. Head pose estimation has two steps to go. Firstly the system makes user's 3D facial model from the first 50 frames. After that, we estimate user's head pose using the facial model. We determined the number 50 frames empirically.

A. Generating 3D Face Model

User's 3D facial model is reconstructed in Structure from Motion method. We input the 50 frames' facial feature points sets into measurement matrix $\boldsymbol{W}$ (1). $\boldsymbol{x}_j^{(i)}$ is jth facial feature point's 2D coordinate of ith frame. $\overline{\boldsymbol{x}}^{(i)}$ is the median point of ith frame. Then we decompose W to two matrices(2). Matrix M represents each frame's rotation and translation. Matrix S is common factor among frames, and this is the 3D facial model we want to obtain.

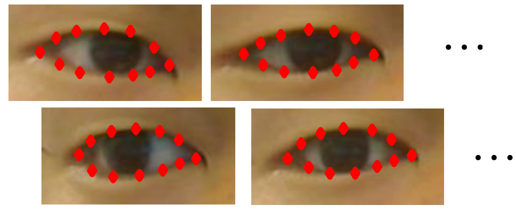


Fig. 2 Various Eye Images including Open/Close and shape variation among persons

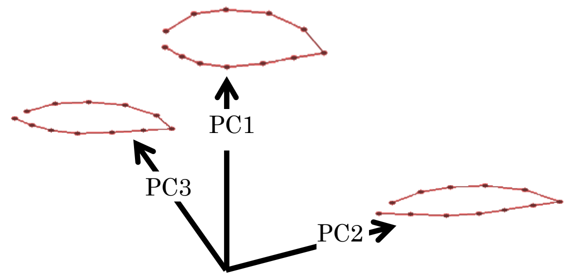


Fig. 3 Eyelid model from principal component analysis

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{x}_1^{(1)} - \overline{\boldsymbol{x}}^{(1)} & \cdots & \boldsymbol{x}_M^{(1)} - \overline{\boldsymbol{x}}^{(1)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x}_1^{(N)} - \overline{\boldsymbol{x}}^{(N)} & \cdots & \boldsymbol{x}_M^{(N)} - \overline{\boldsymbol{x}}^{(N)} \end{bmatrix} \quad (1)$$

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{MS} \\ &= \begin{bmatrix} \boldsymbol{m}^{(1)} \\ \vdots \\ \boldsymbol{m}^{(N)} \end{bmatrix} [\boldsymbol{S}_1 \quad \cdots \quad \boldsymbol{S}_M] \quad (2) \end{aligned}$$

B. Head Pose Estimation

Based on the facial model, we estimate the head pose of new frame (51th frame -). Head pose estimation is to estimate how 3D facial model projects onto current face, so it equal to solve projection matrix $\boldsymbol{P}$ in (3).

$x_j^{(t)}, y_j^{(t)}$ are jth facial feature point's 2D coordinates of tth frame. $\boldsymbol{S}_1 \quad \cdots \quad \boldsymbol{S}_M$ means the facial model. At least 4 facial feature points are required to solve this formula, however, we can solve easily because there are over 60 facial feature points being tracked. By decomposing P in QR method, we can obtain rotation and translation matrices and complete head pose estimation.

$$\begin{bmatrix} x_1^{(t)} & x_2^{(t)} & \cdots & x_M^{(t)} \\ y_1^{(t)} & y_2^{(t)} & \cdots & y_M^{(t)} \end{bmatrix} = \begin{bmatrix} p_{x1}^{(t)} & p_{x2}^{(t)} & p_{x3}^{(t)} & p_{x4}^{(t)} \\ p_{y1}^{(t)} & p_{y2}^{(t)} & p_{y3}^{(t)} & p_{y4}^{(t)} \end{bmatrix} \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{S}_2 & \cdots & \boldsymbol{S}_M \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (3)$$

IV. Eyelid Tracking

In the Iris tracking described in chapter 6, eyelash and hair outside of eyelid become noise. In order to get rid of these noises and focus on the area of iris tracking, we show eyelid tracking method in this section.

It is difficult to extract and track an eyelid without any model because human's eyelid shape is very complex and variable. In proposed method we made a good model which represents human's eyelid shape variation and change. Based on this model, Particle Filter is applied to detect and track an eyelid shape.

A. Eyelid Shape Model

The eyelid shape model was generated as below. Firstly we input eyelid shape by clicking 12 points per one eye. We have repeated this process to a lot of open eye images to learn which include shape variation among persons (Fig. 2). If these eyelid points had no correlation, the model would have 24degree-of-freedom that is 12 points × 2 dimension(x, y), and we would need huge number of particles proportionate to the exponent of degree-of-freedom. In this case, the computational cost increases significantly, so it is not suitable for a real-time tracking system. Secondary therefore we applied Principal Component Analysis to the points of eyelid shapes in order to compress the dimension number of the model. Thanks to this procedure we obtained low-dimensional model which can represent the shape change caused by gaze direction move, expression change, variation among individual. The eyelid model has 5 parameters that are 3 principal components of shape variation (PC1, PC2, PC3), scale (S), eyelid open/close parameter (d). In this paper, we used 250 eye images (5 men and 1 woman) and obtained three dimensional eyelid shape model (Fig. 3).

B. Eyelid Tracking using Particle Filter

In this Eyelid Tracking procedure, we use Particle Filter. (1) It spreads thousands of particles on the input eye image based on the eyelid shape model, and (2) computes each particle's likelihood. Finally (3) the suitable eyelid shape is estimated by calculating weighted mean among all particles. Each process is described in detail below.

(1) Spreading Particles

$$\mathbf{x}_t = (x_t\ y_t\ PC_1\ PC_2\ PC_3\ \mathrm{S}\ \mathrm{d})^{\mathrm{T}}$$
$$\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{v}_t \quad \cdots\cdots\cdots (4)$$

$\mathbf{x}_t$: state vector

$x_t\ y_t$: position

S: scale

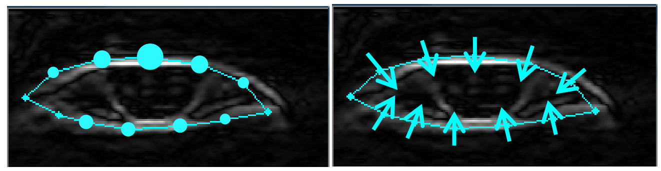
PC_n: each principal component of eyelid model

d: Open/Close ($0 \leq \mathrm{d} \leq 1$)

$\mathbf{v}_t$: noise

The state of eyelid is determined by 7 dimensions (5 dimensions of eyelid shape model and 2 dimensions of its position(x,y)). Actually, the eye position obtained by FaceTracker has a certain margin of error, so it doesn't have enough accuracy for gaze estimation. Therefore we put positional components in order to track more accurately. Although we normalize the eye image size depending on the scale of FaceTracker, we put scale component because eye size ratio to the face differs from person to person.

We spread 1000 particles of various eyelid shapes based on the model and the previous state of eye $\mathbf{x}_{t-1}$. $\mathbf{v}_t$ means adding normal distribution noise.



(a) Edge intensity (b) Edge gradient

Fig.4. Likelihood evaluation

(2) Calculate Likelihood

Next, we calculate the likelihood of each particle. The likelihoods of eye corners are calculated from the return value of SVM classifier, while likelihoods of other eye arcs are calculated focusing on edge. The calculating formula is showed in (5). C_1, C_2 indicate weight constant.

$$\mathrm{L} = C_1 \cdot R_{clm} + C_2(E_{int} + E_{grad}) \cdots\cdots\cdots\cdots\cdots\cdots (5)$$

1. Eye Corners

For inner and outer eye corners we consider their likelihoods as the return value of pre-learned classifier. The classifier was made by SVM, when their features are Local Binary Patterns of patch image of each eye corner. The value is $R_{clm} = 1$ if the classifier determine true, or $R_{clm} = 0$ if false.

2. Eye Arc

For other 10 points on eye arc we consider their edge similarity as their likelihoods. Because eyelid has strong vertical and weak horizontal edges, when iris has weak vertical and strong horizontal edges, the likelihood of eyelid is determined by the sum of vertical edge intensity E_{int} and similarity of edge gradient direction E_{grad}.

Edge intensity is computed in (6). Arc indicates the points on eyelid arc, S(x, y) means each point of the eyelid model. $D(x, y)$ indicates vertical edge intensity on (x, y) (Fig.4(a)).

$$E_{int} = \int_{S(x,y)\,\in arc} D(x,y)ds \cdots\cdots\cdots\cdots\cdots\cdots (6)$$

Similarity of Edge Gradient Direction

Edge gradient similarity is computed in (7). $d_I(x, y)$ indicates the degree of edge gradient direction of input eye image and $d_{arc}(x, y)$ is the degree edge gradient direction on (x, y) of the eyelid model. E_{grad} becomes bigger as the direction similarity of eye image and eyelid model increases (Fig.4(b)).

$$E_{grad} = 1 - \int_{S(x,y)\,\in arc} |d_I(x,y) - d_{arc}(x,y)|ds \cdots (7)$$

(3) Estimate the State

Finally, the suitable eyelid shape is estimated by calculating weighted mean among all particles. We achieve eyelid tracking by repeating these procedures: spreading particles based on current state, calculating likelihood of each particle and estimating the state from all particles.

V. Iris Tracking

Next, we track iris on the same eye image of chapter 4. Iris tracking has 2 step: primary tracking of template matching and secondary refinement of Particle Filter, using 3D eye ball model. Thanks to this 2 step tracking, we can reduce the searching area of Particle Filter so much, which reduce the computation cost. In addition, limiting iris searching area inside eyelid known in chapter 4, we can eliminate outer inlier edges and achieve robust tracking.

A. Eye Ball Model

We consider human's eye as a 3D sphere rotate with its gaze direction, we determine eye ball model whose radius is R and has black disk as iris whose radius is D. The black disk slides on the surface of eye ball along with the gaze direction. Its rotation angle is described by *yaw* and *pitch*(Fig.5. By this model, we can know the relationship(8) between eye ball rotation angle and theoretical eye image including iris projected by this model, and we track iris under the condition of human eye's anatomical structure. Eye ball radius R and iris radius D are determined by the scale of user's face.

$$X = -R \times \cos(\text{pitch}) \times \sin(\text{yaw})$$
$$Y = R \times \sin(pitch) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (8)$$

B. Primary Tracking

In this step we determine yaw and pitch roughly using template matching. We binarize input eye region image, applying p-tile method, whose threshold is determined as 15% of whole pixel become black. Thanks to this process, we can divide black eye region(iris) and while eye region. Then we apply template matching using a black disk template image that represents the iris appearance. Binarized image has shadow of upper eyelid or hair, but by template matching we can focus on iris region rightly. The iris center coordinate in eye image searched by template matching is X_{base}, Y_{base},and we can know the base rotation angle yaw_{base}, $pitch_{base}$ from (8).

C. Secondary Refinement

Based on eye ball model, we track iris accurately with Particle Filter.

(1) Spreading Particles

$$\begin{aligned}\mathbf{x}^t &= \mathbf{x}^t_{base} + \mathbf{x}^t_{diff} \\ &= (yaw_{base}\ pitch_{base})^{\mathrm{T}} + (yaw_{diff}\ pitch_{diff})^{\mathrm{T}} \\ &= \mathbf{x}^{t-1}_{base} + \mathbf{x}^{t-1}_{diff} + \mathbf{v}^{t-1}_{diff}\end{aligned}$$

$\mathbf{x}^t$: state vector

$\mathbf{v}^t$: noise

The state of iris $\mathbf{x}^t$ is the ellipse on eye image determined 2 rotation angle parameters yaw and pitch. yaw_{diff}, $pitch_{diff}$ are the differential angle between yaw_{base}, $pitch_{base}$ estimated in section B and true rotation angle *yaw*, *pitch*. Random sampling can be done in very small yaw_{diff}, $pitch_{diff}$ range, this contributes to reduce search area. Fig.6. show some sample particles.

(2) Calculate Likelihood and Estimate State

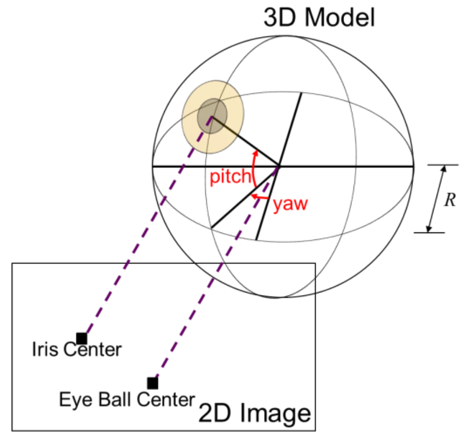


Fig.5. Eyeball Model

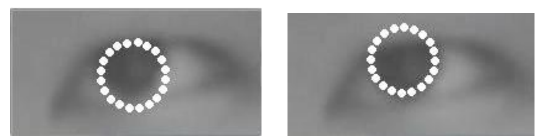


Fig.6. Example of Particle

On each particle spread the system calculates its likelihood. While likelihood calculation of eyelid in chapter 4 its vertical edges are focused, iris's horizontal edge is strong and vertical edge is invisible because of eyelid. So likelihood calculation of iris is done about horizontal edge. For computation reason, we pick up 120 sample points on the ellipse, then we sum up their edge intensity. At last, we calculate weight mean of all particles and estimate suitable yaw and pitch.

VI. Experiment

A. Eyelid Tracking Result

We tested our method on several subjects captured in a room by an ordinary web camera(640x480). Comparing estimated eyelid shape and actual one visibly, we consider true if over 9 points out of 12 points are within 3 pixels. The result is shown in Table.1. Some results of the subjects are shown in Fig. 7. Green line represent head pose axes and yellow ellipse represent iris tracking result of proposal method. The blue one is of previous methods[8]. These images includes eyelid tracking results.

B. Iris Tracking Result

About iris tracking, we tested in FGnet database. The database has 5000 frames of man talking or smiling naturally. Each frame has 720x576pixels and 68 facial feature truth points including iris centers. In this dataset, the man change

Table 1. Eyelid Tracking

Subject	Total frame number	Success Rate[%] Eyelid Tracking
1	423	92.1
2	565	89.6
3	334	88.2
4	465	92.9

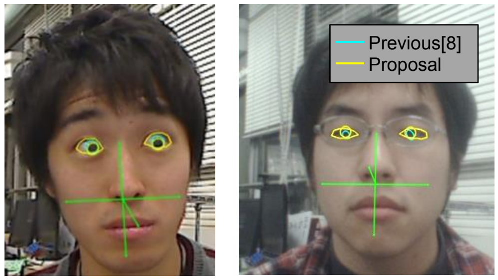


Fig.7. Result

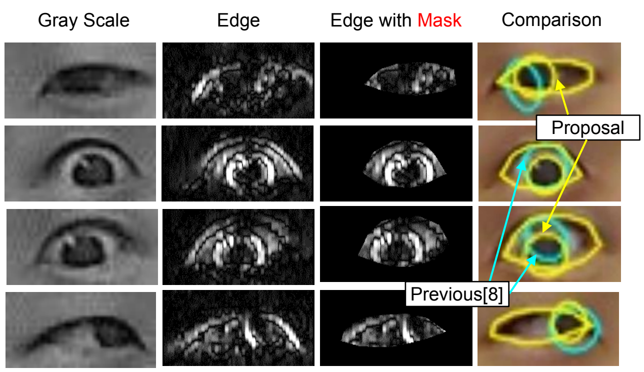


Fig.8. Comparison

his head pose and expression. We compared some methods on this dataset.

Previous method[7] is considered Particle Filter without head pose change, previous method[8] is considered ellipse fitting with RANSAC. Proposal method marked the average error 2.01 [pixel], whereas previous[7] was [2.10] and previous[8] was 2.31 [pixel].

Fig. 8 shows another iris tracking result of the movie capture in room with an ordinary camera(640x480pixel). The edge column's images are the horizontal edges of eye region, which have strong outlier edges like eye corners. However, we can eliminate these edges thanks to the eyelid tracking result (Edge with Mask column). In the comparison column, blue ellipse is the result of previous[8] and blue one is of the proposal method.

VII. Discussion

Our proposal method can track eyelid and iris accurately without any personal calibration. On eyelid tracking, simple and efficient eyelid shape model deal with various shape. In addition, applicable projection transform of eye region image considering head pose gives robust tracking against head pose changes. Different eyelid feature estimation like edge base and classifier contributes to more accurate tracking. Under specific case that black subject or subject who has long hair, our edge base method made some error. On iris tracking we use geometric eye ball model. Primary template matching reduces searching area much, this gives computation cost. The radius parameters are constant as of now; however, we have to optimize these parameters without calibration.

Though our method has various process, each reduction and optimization of computation and multi-threading gives 50fps processing frame rate on a note PC.

References

[1] Dan Witzner Hansen and Qiang JI : "In the Eye of the Beholder:A Survey of Models for Eyes and Gaze", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol.32, No.3 p.478-500 (2010)

[2] Arantxa Villanueva and Rafael Cabeza : "A Novel Gaze Estimation System With One Calibration Point", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART B: CYBERNETICS, Vol.38, No.4 pp.1123-1138 (2008)

[3] H.MORI et al: "A Linear Regression based Gaze Estimation Method for Wide-view Parallax-free Eye-Mark Recorder", Japan Joint Automatic Control Conference, Vol.52, pp. ROMBUNNO.F6-1 (2009)

[4] Y.SUGANO, Y.MATSUSHITA, and Y.SATO : "Unconstrained gaze estimation with learning from mouse operations", MIRU2009 pp.266-273 (2009)(in Japanese)

[5] Y.SUGANO, Y.MATSUSHITA, and Y.SATO : "Gaze estimation using visual saliency", MIRU2010 pp.1507-1514 (2010)(in Japanese)

[6] H.YAMAZOE, A.UTSUMI, T.YONEZAWA, S.AVE : :"Automatic Calibration of 3D Eye Model for Signle-Camera Based Gaze Estimation" IEICE TRANSACTIONS on Information and Systems Vol.J94-D, No.6 pp.998-1006 (2011)

[7] Y.KITAGAWA, H.WU, T.WADA, T.KATO : "On eye-model personalization for automatic visual line estamation", PRMU2007, Vol.106, No.469 pp.55-60 (2007)

[8] Zhang, Wen, Tai-Ning Zhang, and Sheng-Jiang Chang : "Eye gaze estimation from the elliptical features of one iris." Optical Engineering 50(4), 047003 (2011)

[9] Jason M.Saragih, Simon Lucey and Jeffrey F.Cohn : "Deformable Model Fitting by Regularized Landmark Mean-Shift", International Journal of Computer Vision(IJCV), Vol.91, pp.200-215 (2010)

[10] T.Cootes, Talking face video, www-prima.inrialpes.fr/FGnet2002, (FGnet-IST-2000-26434)