Gaze Direction Estimation with a Single Camera Based on Four Reference Points and Three Calibration Images

Shinjiro Kawato, Akira Utsumi, and Shinji Abe

ATR Intelligent Robotics and Communication Laboratories,
Keihanna Science City, Kyoto 619-0288, Japan
{skawato, utsumi, sabe}@atr.jp

Abstract. We propose a method to estimate gaze direction in real time with a single camera based on four reference points and three calibration images. First, the position at which the eyeball center is projected is calculated as a linear combination of those of the reference points. Then, the gaze direction is estimated as a vector connecting the calculated eyeball center and the detected iris center. The algorithm is head pose free. We implemented the algorithm on a PC with a Xeon 2.2-GHz CPU, which works at a rate of 30 fps.

1 Introduction

Gaze estimation is one of the key technologies for human-computer interaction systems. A good review of recent advancements on this topic is presented in [1]. In this paper, we will propose a vision-based practical method for gaze estimation that uses a single camera.

Among various gaze tracking systems, intrusive methods, including head-mounted types, are in general more accurate than remote ones[1]. However, they are troublesome and impose a burden on users; therefore, non-intrusive methods are preferable.

Non-intrusive methods are classified into two categories: active and passive. Active methods use controlled lighting, usually infrared (IR) LEDs, for two different purposes. One is to detect pupils robustly. On-axis and off-axis lighting respectively produce bright-pupil and dark-pupil images[1], and the difference between the images enables robust pupil detection. The other is to make a glint or reflection of the LED on the cornea. The gaze can then be estimated based on the glint position and the center of the pupil.

The glint, however, is a very small spot, and thus an image of high resolution is required to detect it. This means the eye almost fills the screen. Consequently, not only does the focusing depth of field become very shallow, but also a slight movement of the head causes a large displacement in the image. This means the eye can easily fall out of the field of view, thus making it difficult to track. Some systems incorporate an extra wide-angle camera to track the eye and control the

P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3851, pp. 419–428, 2006.

pan and tilt angle of the gaze camera [2][3], though this makes such systems very complicated.

One of the constraints present when using this type of system is that the distance between the user and the camera (and the LEDs) should be very short because of the limited LED power. Usually, users are supposed to be sitting in front of the system within one meter from it.

Among passive methods, systems featuring binocular stereo architectures have a similar constraint. A system described in [4] tracks not only irises but also other predefined feature points such as eye corners and mouth corners. Furthermore, the location of eyeball centers in 3D space are calculated from these feature points calibrated in advance. The gaze direction then is estimated as a line in 3D space connecting an eyeball center and the center of an iris. It works very well. However, in a binocular stereo system, a face should be within a region visible from both cameras. Therefore, the distance from the cameras to the user is very limited with respect to an appropriate image resolution and a base line. A similar system is reported in [5] that uses artificial marks as tracking features and a bright/dark pupil imaging technique.

Here, we propose a single-camera method that allows the use of long shot images after appropriate zooming-up if necessary.

As for single-camera methods, some neural network approaches have been proposed[6][7] that results in very fast calculation. However, it is pointed out that the trained neural network is too sensitive to changes in users, lighting conditions, and even changes within the user[8].

Another single-camera approach is the so-called "circle algorithm." If two circles on parallel planes are observed as ellipses, the normal direction of the support planes can be determined uniquely (two-circle algorithm)[9][10]. Therefore, if the irises are assumed to be circles and both of their images are extracted as ellipses, the gaze direction can be calculated. If we have only one iris image or one ellipse, there will be two possible solutions for the normal direction. Even in such a case, the true solution can also be selected using other cues (one-circle algorithm)[11].

The circle-algorithm approach is very attractive because no calibration is required beforehand. However, it is very difficult to develop a system with current video rate imaging techniques in terms of image resolution. Consider a case when an iris gazing at the camera is a circle in the image with a diameter of 100 pixels. When the eyeball turns ten degrees, the circle changes to an ellipse with the minor axis of 98.5 pixels (see $\cos 10^\circ = 0.9848$) while the major axis remains at 100 pixels. It is quite difficult to detect such small changes stably and accurately.

On the other hand, iris displacement is much greater in the same situation. An eye model used in the simulation in [11] is such that the ratio of the radius of the eyeball to the radius of the iris is 2. According to this model, when the diameter of the iris is 100 pixels, the radius of the eyeball is also 100 pixels. Then, 10-degree rotation of the eyeball results in iris displacement of about 17 pixels (see $\sin 10^\circ = 0.1736$), which seems to be easier to detect than a 1.5-pixel change.

In [8], iris displacement from the inner eye corner is measured to estimate the gaze direction. However, when the face rotates, while the gaze direction is fixed, the iris position relative to the eye corner changes. Therefore, the face direction should be fixed as it is in the calibration processes.

To overcome this problem a special reference point has been proposed in [12]. It is the middle point between the centers of the right and left eyeballs, called the virtual eyeball center. Cleverly, Miyake et al. put two marks on the face where the line connecting two eyeball centers intersects with the surface of the face, and assumed that the middle point of them in the image is the virtual eyeball center. They also detected both of the irises and calculated their middle point. This is called the virtual iris center. The line connecting the virtual eyeball center and the virtual iris center determines the gaze direction.

This idea makes the system head pose free, because the relative position of the virtual eyeball center and the virtual iris center does not change while the gaze is fixed, even when the face rotates. However, one of the marks placed on both sides of the face is likely to be occluded by the face itself when the face turns about twenty degrees or so. Consequently, the system cannot take full advantage of the head pose free algorithm.

Here, we propose another head pose free algorithm in which we use four marks instead of two. However, the constraint on the positions of them is far less restrictive than that used in [12]. Therefore, not only we can place them to be visible for a wide range of head poses, but also we have the scope to replace them with natural image feature points on faces in the future.

We calculate the position of the eyeball center by a linear combination of the positions of the four marks, detect the iris center, and estimate the gaze direction as a line connecting the calculated eyeball center and the detected iris center.

In the next section, we explain the principle of calculating of the fifth point position from the four reference points, and in Section 3, how the principle can be applied to gaze estimation. In Section 4, we briefly describe the image processing technique used in the experiment, and present some experimental results in Section 5. Section 6 concludes the paper.

2 Estimation of the Fifth Point Position

We express a point in 3D space as a vector $\boldsymbol{X}_i = (x_i, y_i, z_i)^T$. When we select four points $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$ on a 3D object such that not all of them are on a plane, the vectors $(\boldsymbol{X}_1 - \boldsymbol{X}_0)$, $(\boldsymbol{X}_2 - \boldsymbol{X}_0)$, and $(\boldsymbol{X}_3 - \boldsymbol{X}_0)$ are linearly independent. Then, for any arbitrary point $\boldsymbol{X}_c$ on the object, there exist α, β, and γ such that

$$(\boldsymbol{X}_c - \boldsymbol{X}_0) = \alpha(\boldsymbol{X}_1 - \boldsymbol{X}_0) + \beta(\boldsymbol{X}_2 - \boldsymbol{X}_0) + \gamma(\boldsymbol{X}_3 - \boldsymbol{X}_0). \tag{1}$$

Because only relative vectors from $\boldsymbol{X}_0$ appear in Eq. (1), selecting the origin of the coordinate as well as its pose makes no difference. For convenience, hereafter, we assume the origin is at the center of gravity, and consider only the rotation of the object.

Our camera model here is the orthogonal projection model. It means that a point $(x, y, z)^T$ in 3D space is projected to $(x, y)^T$ on the image plane. It is known that the modeling error is small when the depth of the object is sufficiently small compared to the distance between the camera and the object.

From Eq. (1), for the image of the object in arbitrary pose,

$$\begin{pmatrix} x_c - x_0 \\ y_c - y_0 \end{pmatrix} = \alpha \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} + \beta \begin{pmatrix} x_2 - x_0 \\ y_2 - y_0 \end{pmatrix} + \gamma \begin{pmatrix} x_3 - x_0 \\ y_3 - y_0 \end{pmatrix} \tag{2}$$

is always satisfied.

When a rotation, expressed by a rotation matrix $\boldsymbol{R}^k$, is applied to the object, a point$\boldsymbol{X}_i$ moves to $(x_i^k, y_i^k, z_i^k)^T = \boldsymbol{R}^k(x_i, y_i, z_i)^T$. Then, from Eq. (2), observed points on the image after rotations $\boldsymbol{R}^0$, $\boldsymbol{R}^1$, and $\boldsymbol{R}^2$ satisfy the following equation.

$$\begin{pmatrix} x_c^0 - x_0^0 \\ x_c^1 - x_0^1 \\ x_c^2 - x_0^2 \\ y_c^0 - y_0^0 \\ y_c^1 - y_0^1 \\ y_c^2 - y_0^2 \end{pmatrix} = \begin{pmatrix} x_1^0 - x_0^0 & x_2^0 - x_0^0 & x_3^0 - x_0^0 \\ x_1^1 - x_0^1 & x_2^1 - x_0^1 & x_3^1 - x_0^1 \\ x_1^2 - x_0^2 & x_2^2 - x_0^2 & x_3^2 - x_0^2 \\ y_1^0 - y_0^0 & y_2^0 - y_0^0 & y_3^0 - y_0^0 \\ y_1^1 - y_0^1 & y_2^1 - y_0^1 & y_3^1 - y_0^1 \\ y_1^2 - y_0^2 & y_2^2 - y_0^2 & y_3^2 - y_0^2 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \tag{3}$$

First, we solve Eq. (3) and obtain the values of α, β, and γ. Then, for an arbitrary rotation of the object even if the point $\boldsymbol{X}_c$ is not observed in the image, its projection point can be calculated from the coordinates $(x_0^k, y_0^k)^T$, $(x_1^k, y_1^k)^T$, $(x_2^k, y_2^k)^T$, and $(x_3^k, y_3^k)^T$ of observed points $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$ using Eq. (2) as follows,:

$$\begin{pmatrix} x_c^k \\ y_c^k \end{pmatrix} = \alpha \begin{pmatrix} x_1^k - x_0^k \\ y_1^k - y_0^k \end{pmatrix} + \beta \begin{pmatrix} x_2^k - x_0^k \\ y_2^k - y_0^k \end{pmatrix} + \gamma \begin{pmatrix} x_3^k - x_0^k \\ y_3^k - y_0^k \end{pmatrix} + \begin{pmatrix} x_0^k \\ y_0^k \end{pmatrix}. \tag{4}$$

For Eq. (3) to be solvable, not all the axes of rotations $\boldsymbol{R}^0$, $\boldsymbol{R}^1$, and $\boldsymbol{R}^2$ should be parallel. Since Eq. (3) is over-constrained, we can solve it by the least-squares method.

3 Gaze Estimation

We assume the gaze direction is a vector from the eyeball center to the iris center. Since the iris is observable, we can calculate its center on the image; on the other hand, the eyeball center is not observable. Therefore, we calculate its position from the observable reference points $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$ as a linear combination of them as mentioned in the previous section.

The gaze direction in the image plane is a vector from the calculated eyeball center to the detected iris center. The angle θ between the gaze direction and the normal of the image plane is calculated as follows, where r is the radius of the eyeball and d is the distance between the calculated eyeball center and the detected iris center in the image (Fig.1).

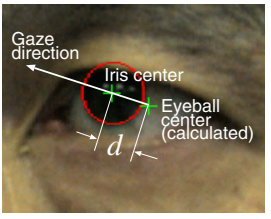


Fig. 1. Gaze direction model

$$\theta = \sin^{-1}(\frac{d}{r}) \tag{5}$$

As for the value of r, an anatomical model can be used like in [11], or we can acquire it from other calibration mean.

The eyeball center corresponds to $\boldsymbol{X}_c$ in the previous section. Although of course it is not observable, in order to calculate α, β, and γ, its projection point should be known in the face images with rotations of $\boldsymbol{R}^0$, $\boldsymbol{R}^1$, and $\boldsymbol{R}^2$. But how? Well, we consider a special case in which we can observe the eyeball center in the image.

In our gaze model, the gaze line is a line in 3D space connecting an eyeball center and the center of an iris. When we gaze at the camera lens, the three points of the center of the lens, the center of the iris, and the eyeball center align. Then, on the image, the center of the iris and the eyeball center are projected at the same point. In other words, in such a special case, the eyeball center is observed as the center of the iris.

In summary, the gaze estimation process is as follows.
(1) Place four reference marks around an eye as $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$. They should not be on the same plane.
(2) Take three images, including the eye and four marks with different head poses while looking at the camera. (Calibration images)
(3) Extract the positions of $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$, and the center of the iris as $\boldsymbol{X}_c$. Now, we have Eq. (3).
(4) Solve Eq. (3) to acquire α, β, and γ.
(5) For an arbitrary gaze and head pose image, extract the positions of $\boldsymbol{X}_0$, $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$, and the center of the iris.
(6) Calculate the projection point of the eyeball center using Eq. (4).
(7) The gaze direction is estimated as a line connecting the calculated eyeball center and the extracted iris center. The angle of the gaze line from the normal of the image plane is calculated by Eq. (5).

4 Experimental System

We developed a simple experimental system to examine the validity of our algorithm. For eye detection and tracking, we use a commercial software[13]. This software library returns locations of both eyes. However, this does not mean the

Fig. 2. Camera setup

iris locations, just dark regions. Therefore, we have to develop an iris detection process. This commercial software detects and tracks eyes under the condition that both are visible. Thus, we take the same approach as in [12], i.e. instead of using a single iris, we use the virtual iris center, which is the middle point of the centers of the right and left irises. Equation (4) then calculates the virtual eyeball center's location..

The camera is of IEEE 1394 interface with an image resolution of 640×480 pixels. The focal length of the lens is $f = 16mm$. The camera is placed on top of the display monitor, with the subject sitting about 90 cm from it. Figure 2 shows the camera setup and the relative face scale in the image.

Figure 3 shows a frame of reference marks. The one and only constraint on the alignment of marks is that not all of them can be on a same plane. The mark is designed for easy detection: it is a white disc 6 mm in diameter with a black circle 3 mm in diameter at the center. The size of the frame is about the distance between the eyes, and it is attachable to the nose part of the glasses so as not to distract the eyes. The mark in the upper-left of the figure is about 17 mm above the plane defined by the other three marks. Figure 4 shows a view of the marks attached to the face. When eye locations are extracted, the region where each mark can exist becomes predictable (see Fig. 4). Each mark is searched in such a region with a simple template matching technique.

The software library we used for eye detection and tracking returns position data where the average gray level of a certain square is lowest in the eye regions.

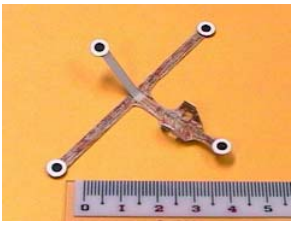


Fig. 3. A frame of reference marks

Fig. 4. Reference marks attached to the face

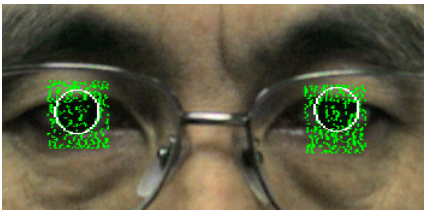


Fig. 5. Example of iris extraction

If the eyes are open, we can expect those positions to be on the irises. However, they are not the center of the irises. Therefore, the center of irises should be searched in a more precise manner.

In many previous works[9][12][10][11], the iris image is binarized, and regions not likely to be iris are eliminated by some means, and then an ellipse is fitted to the edges of the remaining region to extract the center of the iris. However, when binarizing an image, determining an appropriate threshold is almost always a difficult problem. We, therefore, take a different approach. We apply a Laplacian filter and extract zero-cross points as edges in a small region where an iris is likely to exist. We then apply a Hough transform technique for circles for those very many iris edge candidates in order to extract an iris as a circle.

Because the upper and lower parts of the iris are likely to be hidden by the eyelids, only vertical edges are extracted. Consequently, a one-dimensional (horizontal) Laplacian filter is applicable. We can expect that the center of the iris search region is on the iris. Thus, at a zero-cross point, the gradient direction is also taken into account so that the inside is the dark side. In applying the Hough transform for circles, voting to the upper part and lower part (over 60 degrees from the horizontal line) of a circle is suppressed, because that part of the iris is likely to be hidden by the eyelids.

Figure 5 shows an example of iris detection. The two circles are iris locations determined by the Hough transform, while the noisy dots are edge pixels extracted as Laplacian zero-crosses. There are many edges even in the iris region because of reflections of white papers, windows, light sources, etc.

5 Experiment

5.1 Calibration

For Eq. (3), we require three calibration images. The head poses in the calibration images should be different from each other, and the gaze should be directed to the camera. To satisfy these conditions with simple instructions, we showed a target mark at the center of the screen, and asked the subject: for each reference mark except the top one, (1) fit the mark on the target mark by adjusting the head pose; and (2) look at the camera; then (3) press the button. Image of the subject on the screen was flipped horizontally, so that the subject felt it was a mirror. This made it easier for the subject to feed back the image as a head pose

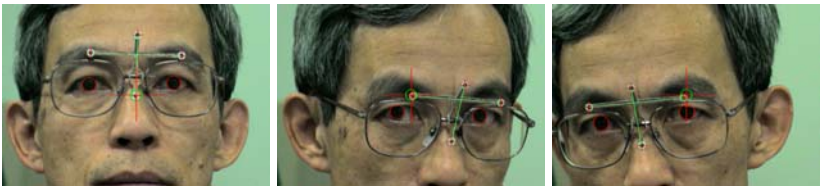
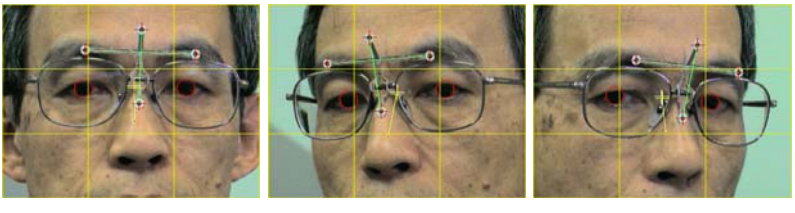


Fig. 6. A set of calibration images. (peripherally clipped)

adjustment, because people are familiar to their own mirror images. Figure 6 shows an example of a set of calibration images.

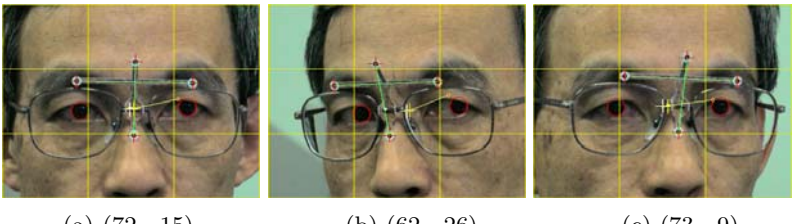
5.2 Head Pose Independency

Figure 7 shows gaze vectors in different head poses while the subject is gazing at the center of the monitor display. The monitor screen has a 5x5 grid, and the image in Fig. 7 is its 3x3 middle component. The two "+" marks are the calculated virtual eyeball center and the detected virtual iris center. The direction of the gaze vector is from the former to the latter, and its magnitude shown here is a summation of the results for the last 15 frames (0.5 seconds). Actual numbers of them in pixels are shown below each. Notice the origin of the coordinate is in the upper-left corner of the input image. Since the camera is set up on top of the monitor display, the estimated gaze direction is downward when the subject



(a) (0, 51) (b) (-12, 66) (c) (5, 54)

Fig. 7. Estimated gaze directions with different head poses while looking at the center of the monitor



(a) (72, -15) (b) (62, -26) (c) (73, -9)

Fig. 8. Estimated gaze directions with different head poses while looking at the upper-right corner of the bezel

is looking at the center of the monitor. The faces in Fig. 7(b) and (c) turn right and left so much that one of the marks used in [12] will be hidden. Nevertheless, the estimated gaze vectors in the three cases are very similar. Actually, figures of gaze directions show that the differences between each of them in both the x-direction and the y-direction are within ± 15 pixels (± 1 pixel when averaging frames).

Figure 8 shows the cases when the subject is looking at the upper right corner of the bezel. Even in these cases, the results show the head pose independency of our method; figures of gaze directions show that the differences between each of them in both x-direction and y-direction is within ± 15 pixels.

5.3 Discussion

Although we cannot observe the location of the eyeball center, the results shown in Figs. 7 and 8 demonstrate that our algorithm to calculate its projected points from the four reference points as described in Section 2 works well.

The estimated gaze directions in Fig. 7(b) and (c) are slightly different. We noticed in experiments that there was a tendency in the drift of estimated gaze direction according to face orientation, and we think this comes from the camera modeling error. However, we employed the orthogonal projection model instead of the perspective model, which made our algorithm simple and robust.

The image resolution we used was rather coarse: the diameter of an iris was about 27 pixels, leading to the distance between the projected points of the eyeball center and the iris center being very short. Consequently, mainly due to fluctuations of the video signal, the estimated gaze direction fluctuates frame by frame. Therefore, in our experiment, we had to employ time averaging (0.5 seconds or 15 frames) to attain stable results. Consequently, the system contained a time delay, even though it processed 30 frames per second.

6 Conclusions

We proposed a method to estimate the gaze direction using a single camera. The position of the eyeball center is calculated by a linear combination of four reference points. To determine the coefficients of the linear combination, we need three calibration images with different head poses. The gaze direction is estimated as a vector from the calculated eyeball center to the detected iris center. We demonstrated the validity of the algorithm in experiments. The algorithm is head pose independent; or in other words, head pose is determined with respect to four reference points. The system, implemented on a PC with Xeon 2.2-GHz processor, could process 30 frames per second. A demonstration video clip can be opened at the author's home page. (http://www.mis.atr.jp/~skawato)

In the prototype system, locations of reference marks and irises are detected with pixel accuracy. However, because the eyeball center and the iris center are very close, one pixel error causes a relatively large direction error. Therefore, detecting their locations at sub-pixel accuracy remains as future work. In the

future, we plan to apply natural feature points extracted on the face, instead of artificial reference marks, to calculate the location of the eyeball center.

This research was supported in part by the National Institute of Information and Communications Technology.

References

1. Morimoto, C.H., Mimica, M.R.M.: Eye gaze tracking techniques for interactive applications. Computer Vision and Image Understanding **98** (2005) 4–24
2. Ohno, T., Mukawa, N., Kawato, S.: Just blink your eyes: A head-free gaze tracking system. Proc. of CHI 2003 (2003) 950–951
3. Yoo, D.H., Chung, M.J.: A novel non-intrusive eye gaze estimation using corss-ratio under large head motion. Computer Vision and Image Understanding **98** (2005) 25–51
4. Matsumoto, Y., Zelinsky, A.: An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement. Proc. IEEE 4th Int. Conf. on Automatic Face and Gesture Recognition (2000) 499–504
5. Tomono, A., Kishino, F., Kobayashi, Y.: Pupil exraction processing and gaze point detection system allowing head movement (in japanese). IEICE(D-II) **J76-D-II** (1993) 636–646
6. Baluja, S., Pomerleau, D.: Non-intrusive gaze tracking using artificial neural networks. Technical Report CMU-CS-94-102 (1994)
7. Schiele, B., Waibel, A.: Gaze tracking based on face-color. Proc. Int. Workshop on Automatic Face and Gesture Recognition (1995) 344–349
8. Zhu, J., Yang, J.: Subpixel eye gaze tracking. Proc. Int. Conf. on Automatic Face and Gesture Recognition (2002) 124–129
9. Wang, J.G., Sung, E.: Gaze determination via images of irises. Image and Vision Computing **19** (2001) 891–911
10. Wu, H., Chen, Q., Wada, T.: Conic-based algorithm for visual line estimation from image. Proc. Int. Conf. on Automatic Face and Gesture Recognition (2004) 260–265
11. Wang, J.G., Sung, E., Venkateswarlu, R.: Estimating the eye gaze from one eye. Computer Vision and Image Understanding **98** (2005) 83–103
12. Miyake, T., Haruta, S., Horihata, S.: Image based eye-gaze estimation irrespective of head direction. Proc. IEEE Int. Symposium on Industrial Electronics **1** (2002) 332–336
13. : http://www.red.atr.jp/product/08/pro_08.html. (2004)