Abstract

This paper describes a conic-based algorithm for estimating visual line from a single monocular image. By assuming that the visual lines of both eyes are parallel and the iris boundaries are circles, we propose a “two-circle” algorithm that can estimate the normal vector of the supporting plane of the iris boundaries, from which the visual line is calculated. Our new method does not use either the eye corners, or some heuristic knowledge about the structure of the eye. Another advantage of our algorithm is that a camera with an unknown focal length can be used without assuming the orthographical projection. This is a very useful feature because it allows one to use a zoom lens and to change the zooming factor whenever he or she likes. It also gives one more freedom of the camera setting because keeping the camera far from the eyes is not necessary in our method. The extensive experiments over simulated images and real images demonstrate the robustness and the effectiveness of our method.

1. Introduction

Gaze direction provides several functions such as giving cues of people’s interest, attention or reference by looking at an object or a person. It also plays an important role in many human computer interaction applications. Most of the gaze researches are focused on the eye detection or gaze tracking, which only give the position of the eyes on an image plane[3]-[10]. Many of them assume that the iris contours in the image are circles, which is not always true.

Mastumoto et al assumed that the position of the eyeball center relative to the eye corners is known (manually adjusted), and the iris contour is a circle. They located the eye corners by using a binocular stereo system, and estimated the iris center by applying Hough Transformation to the detected iris contours. Then, they calculate the 3D position of the eyeball center from the head pose and two eye corners. The gaze direction is calculated from the iris center and the eyeball center[6].

Wang et al presented a “one-circle” algorithm for estimating the gaze direction using a monocular camera with known focal length[3]. They detected the elliptical iris contour and used it to calculate the normal vector of the supporting plane of the circular iris boundary. In order to resolve the ambiguity of the multi-solutions, they assumed that the 3D distance between the eyeball center to each of the two eye corners are equal. However, they did not show how the 3D positions of two eye corners could be obtained.

When people look at some where not too near, the visual lines of the both eyes are approximately parallel. In this paper, we present a “two-circle” algorithm for estimating the visual line from two iris contours detected from one monocular image. Our “two-circle” algorithm does not require the whole irises or their centers are viewable, or known radius of them. It also allows one to use a camera with unknown focal length. Compared with the “one-circle” algorithm, our “two-circle” algorithm has two advantages: 1) It can give an unique solution of the visual line without using eye corners, 2) It does not require a known focal length of the camera, and it does not use the orthographical projection approximation. The second feature allows us to forget the orthographical approximation, which does not always stand because eyes are not always far from cameras.

2. The “TWO-CIRCLES” Algorithm

2.1. Elliptical cone and circular cross section

Here, we describe the problem of estimating the direction of a circle plane from one perspective view. M.Dhome[11] addressed it in a research about the pose estimation of an object of revolution. We give a rigorous description here, from which we derive the “two-circle” algorithm. More detailed explanation can be found in the paper [1].

When a circle is projected onto an image plane by perspective projection, it shows an ellipse in general case. Considering a camera coordinate system that the origin is the optical center and the Z-axis is the optical axis, then the oblique elliptical cone defined by the optical center and the ellipse on the image plane $z = -f$ ($f$ is the focal length) can be described by the following equation in quadric form,

$$P^TQP = 0,$$  (1)
where

\[ Q = \begin{pmatrix} A & B & -D \\ B & C & -E \\ -D & -E & F \end{pmatrix}. \]  

(2)

\( Q \) can be expressed by its normalized eigen-vectors \((v_1, v_2, v_3)\) and eigen-values \((\lambda_1, \lambda_2, \lambda_3)\) as following:

\[ Q = V \Lambda V^T, \]

(3)

where

\[ \begin{align*} A &= \text{diag}\{\lambda_1, \lambda_2, \lambda_3\} \\
V &= \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}. \end{align*} \]

(4)

Considering a supporting plane coordinate system that the origin is also the optical center, but the 
\( Z \)-axis is defined by the normal vector of the supporting plane of the circle to be viewed. If the center and the radius of the circle is given by \((x_0, y_0, z_0)\) and \( r \) respectively, the oblique circular cone defined by the optical center and the circle is given by,

\[ P_c^T Q_c P_c = 0, \]

(5)

where

\[ Q_c = \begin{pmatrix} 1 & 0 & -z_0 \\
0 & 1 & -y_0 \\
-\frac{z_0}{z_0} & -\frac{y_0}{z_0} & \frac{z_0^2 + y_0^2 + z_0^2}{z_0^2} \end{pmatrix}. \]

(6)

Since \( Q_c \) and \( Q \) describe the same cone surface, there is a rotation matrix \( R_c \) that transforms \( P_c \) to \( P \) as following,

\[ P = R_c P_c. \]

(7)

Since \( k Q_c \) describes the same cone as of \( Q_c \) for any non-zero \( k \), we obtain the following equation from Eq.(7), Eq.(5), Eq.(3) and Eq.(1),

\[ (V^T R_c)^T A (V^T R_c) = k Q_c. \]

(8)

Because \( VV^T = R_c R_c^T = I \), we have

\[ (V^T R_c) (V^T R_c)^T = I. \]

(9)

Without losing generality, we assume that

\[ \lambda_1 \lambda_2 > 0, \quad \lambda_1 \lambda_3 < 0, \quad |\lambda_1| \geq |\lambda_2|. \]

(10)

Solving Eq.(8) and Eq.(9), we obtain,

\[ V^T R_c = \begin{pmatrix} g \cos \alpha & S_1 g \sin \alpha & S_2 h \\
g \sin \alpha & -S_1 g \cos \alpha & 0 \\
S_1 S_2 h \cos \alpha & S_2 h \sin \alpha & -S_1 g \end{pmatrix}, \]

(11)

where \( \alpha \) is a free variable, \( S_1 \) and \( S_2 \) are undetermined signs, and

\[ g = \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}}, \quad h = \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}}. \]

(12)

Thus \( R_c \) can be calculated as following from Eq.(9):

\[ R_c = V (V^T R_c). \]

(13)

Then the normal vector of the supporting plane can be calculated as following,

\[ N = R_c \begin{pmatrix} 0 \\
0 \\
1 \end{pmatrix} = V \begin{pmatrix} S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\
0 \\
-S_1 \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{pmatrix}. \]

(14)

However, since the two undetermined signs \( S_1 \) and \( S_2 \) are left, we have four possible answers for \( N \). If we define the normal vector of the supporting vector to be the one directing to the camera, we have the constraint, \( N \cdot \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T > 0 \). Then at least one of \( S_1 \) and \( S_2 \) can be determined, thus the number of the possible answers of \( N \) is at most two.

### 2.2. The “TWO-CIRCLES” Algorithm

As described in the section 2.1, the normal vector of the supporting plane of a circle can be determined from one perspective image, when the focal length is known. When the focal length is unknown, according to section 2.1, two oblique elliptical cones can be formed from two circles in parallel planes detected as ellipses if we give a focal length. From each of them, the normal vector of the supporting plane can be estimated independently. Only if we give the correct focal length, the normal vectors estimated from each of the detected ellipses will become parallel.

Let \( \mathbf{N}_i(f); (i = 1, 2) \) denote the normal vector estimated from ellipse \( i (= 1, 2) \). Because the supporting planes of the two circles are parallel, \( \mathbf{N}_1(f) \) and \( \mathbf{N}_2(f) \) are also parallel. This constraint can be expressed by \( \mathbf{N}_1(f) \times \mathbf{N}_2(f) = 0 \). Thus by minimizing the following expression, the normal vector as well as the focal length \( f \) can be determined. The undermined signs remained in Eq.(14) can also be determined at the same time.

\[ (\mathbf{N}_1(f) \times \mathbf{N}_2(f))^2 \rightarrow \min. \]

(15)

Therefore, by taking a face image where two eyes are viewed, two circular iris contours are detected and fitted with ellipses, the normal vector of their supporting planes can be estimated, from which the visual line can be calculated. Here, the iris center, the eyeball center, the iris radius and other facial features except the iris contour are not used.

### 3. Experimental Results

**Experiment With Simulated Images:** We first tested our algorithm on some simulated images of scenes containing coplanar circles. The image resolution is 640 × 480 [pixel].
We set the distance between the optical center and the supporting plane is 3.0 [meter], and the radius is 1.0 [meter]. Two cases of camera setting are shown in top of Table 1. We used 32 images and 17 images containing two ellipses randomly selected from “case-1”, and “case-2”, respectively. The experimental results are summarized in down of Table 1 with suffixes 1 and 2.

### Table 1: Experiment with simulated images

<table>
<thead>
<tr>
<th>Camera setting</th>
<th>case-1</th>
<th>case-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f) (pixel)</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>(\beta) (degree)</td>
<td>10</td>
<td>30</td>
</tr>
<tr>
<td>(\theta) (degree)</td>
<td>40</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Estimation error</th>
<th>RMS error</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f_1) (%)</td>
<td>2.76</td>
<td>4.61</td>
</tr>
<tr>
<td>(\beta_1) (degree)</td>
<td>0.36</td>
<td>0.47</td>
</tr>
<tr>
<td>(\theta_1) (degree)</td>
<td>0.57</td>
<td>0.97</td>
</tr>
<tr>
<td>(f_2) (%)</td>
<td>2.40</td>
<td>3.96</td>
</tr>
<tr>
<td>(\beta_2) (degree)</td>
<td>0.11</td>
<td>0.15</td>
</tr>
<tr>
<td>(\theta_2) (degree)</td>
<td>0.51</td>
<td>0.85</td>
</tr>
</tbody>
</table>

### Experiment With Real Images:

Then, we tested our method with a real image shown in Fig.1(a). The image size is 1600 \(\times\) 1200 [Pixel]. The detected and fitted ellipses are superimposed on the original image. The parameters of the detected ellipses are shown in top of Table 2, where \(a\), \(b\), \(\theta\) and \((x_0, y_0)\) is the major axis, minor axis, the angle between the major axis and the \(X\)-axis, and the center of the ellipse, respectively. The origin of the image coordinate system is the image center, and the \(Y\) axis directs to the upper direction.

The normal vector of the supporting plane (the table) and the focal length of the camera can be estimated using any two of the three circles (CD discs). The results are summarized in down of Table 2. Since the true answer is not available, we used the estimated results to convert the original image to a vertical view of the supporting plane (the table) to see if it resembles the real scene or not. A result obtained by using a circle pair of CD1 and CD2 is shown in Fig.1 (b). In the converted image, each circular object shows a circle and the book shows a rectangle. This indicates that our “two-circle” algorithm could give correct results for real images.

### Experiment With Real Face Images:

Last, we tested our method on many real face images taken by two digital still cameras (Canon EOS DigitalKiss) with a zoom lens \((f=18-55[mm] \text{ or } f=2450-7500[Pixel])\). The image size is 3072 \(\times\) 2048 [Pixel]. Some images used in the experiment are shown in Fig.4. The experimental environment for taking the images No.1-No.6 is shown in Fig.2, where we let a user to look at a marker far away from he/she in the frontal direction. The image No.1-No.2 were taken by camera C1, and No.3-No.6 were taken by camera C2. The images No.7-No.9 were taken in a situation where no instructions about the head pose or the fixation direction were given to users.

For each image, the eye regions are detected[2]. Some simple image enhancement processing are applied to the eye region to make the iris clear. Then the iris contours are detected and fitted with ellipses, which are used to calculate the direction of the visual line with the “two-circles” algorithm. Fig.3 shows the procedure of visual line estimation. Some experimental results of estimated visual line were showed as arrows superimposed on iris in the face image (Fig.4).

### 4. Conclusion

This paper has presented a conic-based algorithm for estimating visual line from a single monocular image. This method only uses the iris contours and does not require the whole iris boundaries viewable, and it does not use the information about the iris centers. Compared with existing method, our method does not use either the eye corners, or the heuristic knowledge about the eyeball. Another advantage of our algorithm is that a camera with an unknown focal length can be used without assuming the orthographical
Figure 2: The experimental environment.

![Diagram showing experimental environment with two circles and a triangle labeled C1 and C2, with measurements of 100 cm and 2.7°.](image)

(a) Original image, (b) Detecting iris, (c) fitted iris contours,

<table>
<thead>
<tr>
<th>Image size (Pixel)</th>
<th>( a_i: b_i: \theta_i: x_{0i}: y_{0i} )</th>
<th>( a_r: b_r: \theta_r: x_{0r}: y_{0r} )</th>
<th>Visual line direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 720 \times 480 )</td>
<td>33: 33: -35.9: 190: 63</td>
<td>-35.0: -0.07: 0.94</td>
<td></td>
</tr>
<tr>
<td>34: 30: 83.3: -178: 56</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(d) Estimated ellipses parameter and visual line direction.

Figure 3: The procedure of the visual line estimation.

projection. In order to obtain satisfy results, the size of the iris contour in the image should be big enough. We consider that this requirement is an acceptable because High-resolution cameras are becoming popular and less expensive nowadays.

Acknowledgments: This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (C), 16500112, 2004.

References