Conic-Based Algorithm for Visual Line Estimation from One Image

Haiyuan WU†, Qian CHEN†, Taro IIO†, and Toshikazu WADA†
† Faculty of Systems Engineering, Wakayama University, Wakayama City, Wakayama, 640-8510, Japan
E-mail: †{wuhy,chen,twada}@sys.wakayama-u.ac.jp, ††iio@vrl.sys.wakayama-u.ac.jp

Abstract This paper describes a novel method to estimate visual line from a single monocular image. By assuming that the visual lines of the 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 direction of the visual line can be calculated. Most existing gaze estimation algorithms require eye corners and some heuristic knowledge about the structure of the eye as well as the iris contours. In contrast to the exiting methods, our one does not use those additional information. 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.

Key words Visual Line Estimation, Conic-Based, Single Monocular Image, Unknown Focal Length

1. Introduction

Gaze direction provides several functions such as giving cues of people’s interest, attention or reference by looking at an object or person. It also plays an important role in many human computer interaction applications. So, the estimation of gaze direction is a very important task.

According to the anatomy, the visual line defines the gaze direction, which is a straight line passing through the object of fixation, the nodal point and the fovea. Because the nodal point and the fovea are in the eyeball, they cannot be observed with a normal video camera. Therefore the visual line cannot be estimated directly. However, since the visual line and the line connecting the eyeball center and the pupil center, or the visual line and the normal vector of the supporting plane of the iris boundary form a (approximately) constant angle, the visual line can be calculated indirectly from the 3D position of the eyeball center and the pupil center, or the normal vector of the supporting plane of the iris boundaries.

Most of the gaze researches are focused on the eye detection or gaze tracking, which only give the position of the eyes on the image plane [1]- [8]. Many of them assume that the iris contours in the image are circles, which is not always true.

Mastumoto et al proposed a gaze estimation method [4]. They assumed that the position of the eyeball center relative to the two 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.

Wang et al presented a “one-circle” algorithm for estimating the gaze direction using a monocular camera with known focal length [1]. 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 the two eye corners were obtained.

There are two kinds of approaches to estimate the visual line. The first one uses elliptical iris contour to estimate the normal vector of the supporting plane of the iris boundary. The other is to estimate the invisible eyeball center, then to use it to calculate the visual line together with the detected iris center.

We consider that the former is a more useful and more reliable approach because it depends on less image features and knowledge about the eyes than the eyeball center based algorithms. The accuracy of the conic-based algorithm depends on the precision of the elliptical iris contour detected from an input image. 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 acceptable because

( 1 ) High-resolution cameras are becoming popular and less expensive nowadays. For example, a HDTV camcorder made by JVC (GR-HD1) has a resolution of 1280x720 pixel, is available at a price of about three thousand US dollars.

( 2 ) We are developing an active camera system for observing human faces, which consists of a camera mounted on a pan-tilt unit. The camera parameters including focal length and the viewing direction are controlled by a computer.

By using it together with a real-time face detecting/tracking software, big face images can be obtained continuously.

In this paper, we present a “two-circle” algorithm for estimating the visual line from the iris contours detected from one monocular image. Our “two-circle” algorithm does not require the whole irises or their centers to be viewable, or the 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 extensive experiments over simulated images and real images demonstrate the robustness and the effectiveness of our method.

2. The “TWO-CIRCLES” Algorithm

When people look at a place that is not very near, the visual lines of the both eyes are approximately parallel, thus the supporting planes of the both iris boundaries are also parallel. Here, we give a “two-circles” algorithm that can estimate the normal vector of the supporting plane(s) from an image of two circles on the same plane or on two different but parallel planes taken by a camera with unknown focal length. Therefore the “two-circles” algorithm can be used to estimate the visual line when the irises of the both eyes have been detected from an input image.

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 [9] 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.

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 ellipse on the image plane \( z = -f \) (\( f \) is the focal length) can be described by the following equation,

\[
Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0. \tag{1}
\]

A bundle of straight lines passing through the optical center and the ellipse forms an oblique elliptical cone that can be described by,

\[
P^TQ = 0, \tag{2}
\]

where

\[
Q = \begin{pmatrix}
A & B & -\frac{D}{f} \\
B & C & -\frac{E}{f} \\
-\frac{D}{f} & -\frac{E}{f} & \frac{F}{f^2}
\end{pmatrix}. \tag{3}
\]

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:

\[
Q = VAV^T, \tag{4}
\]

where

\[
\begin{cases}
\Lambda = \text{diag}\{\lambda_1, \lambda_2, \lambda_3\} \\
V = \begin{pmatrix}
v_1 \\
v_2 \\
v_3
\end{pmatrix}
\end{cases}. \tag{5}
\]

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. Then the supporting plane and the circle can be described by the following expressions,

\[
\begin{cases}
z = z_0 \\
(x - x_0)^2 + (y - y_0)^2 = r^2
\end{cases} \tag{6}
\]

where \((x_0, y_0, z_0)\) is the center and \(r\) is the radius of the circle. A bundle of straight lines passing through the optical center and the circle forms an oblique circular cone described by,

\[
P_c^TQ_cP_c = 0, \tag{7}
\]
Since the oblique circular cone \( Q_c \) and the oblique elliptical cone \( Q \) are the same cone surface, there is a rotation matrix \( R_e \) that transforms \( P_e \) to \( P \) as following.

\[
P = R_e P_e.
\]

(9)

Since \( kQ_e \) describes the same cone as of \( Q_e \) for any non-zero \( k \), we obtain the following equation from Eq.(9), Eq.(7), Eq(4) and Eq.(2),

\[
(V^T R_e)^T \Lambda (V^T R_e) = k Q_e.
\]

(10)

If we use \( R \) to indicate \( V^T R_e \) as following,

\[
R = \begin{pmatrix}
    r_{1x} & r_{2x} & r_{3x} \\
    r_{1y} & r_{2y} & r_{3y} \\
    r_{1z} & r_{2z} & r_{3z}
\end{pmatrix} = V^T R_e,
\]

(11)

we have \( R^T R = I \). By substituting Eq.(11) for \( V^T R_e \) in Eq.(10), we obtain the following equations,

\[
\begin{align*}
\lambda_1 (r_{1x}^2 - r_{2x}^2) + \lambda_2 (r_{1y}^2 - r_{2y}^2) + \lambda_3 (r_{1z}^2 - r_{2z}^2) &= 0 \\
\lambda_1 r_{1x} r_{2x} + \lambda_2 r_{1y} r_{2y} + \lambda_3 r_{1z} r_{2z} &= 0
\end{align*}
\]

(12)

Without losing generality, we assume that \( \lambda_1 \lambda_2 > 0 \), \( \lambda_1, \lambda_3 < 0 \), \( |\lambda_1| \geq |\lambda_2| \).

(13)

Solving Eq.(12) and \( R^T R = I \), we obtain,

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

(14)

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

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

(15)

Thus \( R_e \) can be calculated as following from Eq.(11):

\[
R_e = VR
\]

(16)

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

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

(17)

By substituting Eq.(16) for \( R_e \) in Eq.(10), \( k, x_0/z_0, y_0/z_0 \) and \( r/z_0 \) are determined. Then the center of the circle (denoted by \( C \)) described in the camera coordinate system can be computed by the following expression,

\[
C = z_0 R_e \begin{pmatrix}
    x_0/z_0 \\
    y_0/z_0 \\
    1
\end{pmatrix} = z_0 V \begin{pmatrix}
    S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\
    0 \\
    -S_1 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}}
\end{pmatrix}.
\]

(18)

However, since the two undetermined signs \( S_1 \) and \( S_2 \) are left, we have four possible answers for \( N \) and \( C \). If we define the normal vector of the supporting vector to be the one directing to the camera, and if we know that the center of the circle is in front of the camera, we have the following constraints,

\[
\begin{pmatrix}
    N \cdot (0 0 1)^T > 0 \\
    C \cdot (0 0 1)^T < 0
\end{pmatrix}.
\]

(19)

Then at least one of \( S_1 \) and \( S_2 \) can be determined, thus the number of the possible answers of \( N \) and \( C \) is at most two.

2.2 The “TWO-CIRCLES” Algorithm

As described in the section 2.1, the normal vector of the supporting plane and the center of the circle can be determined from one perspective image, when the focal length is known. When the focal length is unknown, two iris contours are detected and fitted with ellipses, according to section 2.1, two oblique elliptical cones can be formed from the each of the detected 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 \( N_1(f) \) denote the normal vector estimated from one of the two ellipses and \( N_2(f) \) denote the normal vector estimated from the other. Because the supporting planes of the two circles are parallel, \( N_1(f) \) and \( N_2(f) \) are also parallel. This constraint can be expressed by the following equation,

\[
(N_1(f) \times N_2(f)) = 0.
\]

(20)

Thus by minimizing the following expression, the normal vector as well as the focal length \( f \) can be determined. The undetermined signs remained in Eq.(17) and Eq.(18) can also be determined at the same time.

\[
(N_1(f) \times N_2(f))^2 \rightarrow \min.
\]

(21)

Therefore, by taking a face image where two eyes are viewed, the normal vector of the supporting planes of the both circular iris boundaries 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

3.1 Experiment With Simulated Images

We first tested our algorithm on some simulated images to examine the estimation error caused by the quantification error. We used computer graphics to synthesize images of scenes containing coplanar circles. The image resolution is 640 × 480 [pixel].
We first set the focal length to 200 [pixel], the tilt angle $\theta$ (the angle between the optical axis and the supporting plane) to 40 [degree], the roll angle $\beta$ (the rotation angle about the optical axis) to 10 [degree], and the distance between the optical center and the supporting plane to 3.0 [meter]. We called this camera setting as "case-1" hereafter, and used it to synthesize images of circles of which the radius is 1.0 [meter]. Figure 1(a) shows an image containing all the circles used in the experiment using the “case-1” camera setting.

We used 32 images containing two ellipses randomly selected from the image shown in Figure 1(a).

We also used a different camera setting (“case-2”) to do the similar experiment. This time, we set the focal length to 300 [pixel], $\theta = 50$ [degree] and $\beta = 30$ [degree]. All other parameters remained same as “case-1”. Figure 1(b) shows an image containing all the circles used in the experiment of the “case-2” camera setting.

We used 17 images containing two circles randomly placed on the supporting plane, the ones shown in Figure 1(b).

From each image, two ellipses were detected and fitted, which were then used to estimate the normal vector of the supporting plane and the focal length. From the normal vector of the supporting plane, the tilt angle and the roll angle were calculated. The estimated focal length, the tilt angle and the roll angle were compared to the camera parameters used to synthesize the images with CG. The experimental results are summarized in Table 1 with suffixes 1 and 2. We noticed that if the minor axis is greater than 30 pixels, the estimation error will become small enough.

### 3.2 Experiment With Real Images

We tested our method with a real image shown in Figure 2(a). It contains three circular objects: CD1 (the big CD on the table), CD2 (the big CD on the book) and CD3 (the small CD on the table). CD1 and CD3 are coplanar, and CD2 is on another plane that is parallel to the supporting plane of CD1 and CD3. The image size is $1600 \times 1200$[Pixel]. The detected and fitted ellipses are superimposed in the original image. The parameters of the detected ellipses are shown in 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 Table 3. Since

<table>
<thead>
<tr>
<th>No.</th>
<th>CD1</th>
<th>CD2</th>
<th>CD3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$ (pixel)</td>
<td>332</td>
<td>307</td>
<td>242</td>
</tr>
<tr>
<td>$b$ (pixel)</td>
<td>180</td>
<td>127</td>
<td>162</td>
</tr>
<tr>
<td>$\theta$ (degree)</td>
<td>13.2</td>
<td>-1.1</td>
<td>3.5</td>
</tr>
<tr>
<td>$x_0$ (pixel)</td>
<td>400</td>
<td>-440</td>
<td>-162</td>
</tr>
<tr>
<td>$y_0$ (pixel)</td>
<td>-115</td>
<td>325</td>
<td>-252</td>
</tr>
</tbody>
</table>
the true answer is not available, we used the estimation results to convert the original image to a vertical view of the table to see if it resembles the real scene or not. The three results obtained by using different circle pairs are shown in figure 2 (b), (c) and (d). In the converted images, each circular object shows a circle and the book shows a rectangle. This indicates that the “two-circle” algorithm could give correct results for real images.

<table>
<thead>
<tr>
<th>No.</th>
<th>θ (degree)</th>
<th>β (degree)</th>
<th>Normal Vector</th>
<th>f (pixel)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>33.3</td>
<td>4.9</td>
<td>(0.07 0.83 0.55)</td>
<td>1998</td>
</tr>
<tr>
<td>2-3</td>
<td>34.2</td>
<td>6.1</td>
<td>(0.09 0.82 0.56)</td>
<td>1893</td>
</tr>
<tr>
<td>3-1</td>
<td>33.7</td>
<td>6.3</td>
<td>(0.09 0.83 0.55)</td>
<td>2007</td>
</tr>
</tbody>
</table>

3.3 Experiment With Real Face Images

We tested our method by applying it to many real face images taken by two digital still cameras (Canon EOS DigitalKiss) with a zoom lens (f=18-55[mm] or f=2450-7500[Pixel]). The image size is 3072 × 2048[Pixel]. Some of the images used in the experiment are shown in Figure 5. The experimental environment for taking the images No.1-No.6 is shown in figure 3, 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 uncontrolled environment.

![Diagram of test environment](image)

For each image, the eye regions are detected [28]. Some simple image enhancement processings 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 normal vector of the supporting plane of the iris boundaries with the “two-circles” algorithm. The direction of the visual line is obtained from the estimated normal vector. Figure 4 shows the procedure of visual line estimation. The experimental results obtained from the images shown in Figure 5 are summarized in Table 4, and estimated visual line was showed as arrows superimposed on iris in each face image.

![Diagram of visual line estimation](image)

<table>
<thead>
<tr>
<th>Image size (Pixel)</th>
<th>θl: θr: θl': θr': x0l: y0l'</th>
<th>Visual line direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>720 × 480</td>
<td>33: 33: -35.9: 190: 63</td>
<td>(-0.35 -0.07 0.94)</td>
</tr>
</tbody>
</table>
4. Conclusion

This paper has presented a new method to estimate 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 projection. The extensive experiments over simulated images and real images demonstrate the robustness and the effectiveness of our method.

<table>
<thead>
<tr>
<th>No.</th>
<th>Elliptical iris contours</th>
<th>Visual line direction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>left: $a_l$, $b_l$, $\theta_l$, $x_{0l}$, $y_{0l}$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>112: 100: 84.5: 672: 243</td>
<td>(-0.28 -0.00 0.96)</td>
</tr>
<tr>
<td></td>
<td>109: 103: -79.7: 477: 195</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>102: 99: 38.9: 861: 102</td>
<td>(-0.07 0.02 0.99)</td>
</tr>
<tr>
<td></td>
<td>100: 98: -83.4: 146: 84</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>69: 56: -73.9: 390: 291</td>
<td>(-0.57 0.13 0.81)</td>
</tr>
<tr>
<td></td>
<td>63: 52: -79.4: 277: 313</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>68: 63: -89.4: 635: 89</td>
<td>(-0.61 -0.01 0.79)</td>
</tr>
<tr>
<td></td>
<td>74: 59: -85.9: 14: 78</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>88: 75: -76.5: 324: 492</td>
<td>(-0.56 0.02 0.83)</td>
</tr>
<tr>
<td></td>
<td>78: 60: -81.5: -375: 469</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>73: 70: 86.0: 647: 97</td>
<td>(-0.49 -0.00 0.87)</td>
</tr>
<tr>
<td></td>
<td>73: 62: -83.6: -109: 54</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>62: 55: -63.0: -98: -175</td>
<td>(-0.36 0.26 0.89)</td>
</tr>
<tr>
<td></td>
<td>59: 46: -68.4: -588: -160</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>46: 37: -5.3: 107: 79</td>
<td>(-0.13 0.55 0.83)</td>
</tr>
<tr>
<td></td>
<td>47: 35: -26.2: -325: 66</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>35: 33: 62.8: 456: 53</td>
<td>(0.13 0.20 0.97)</td>
</tr>
<tr>
<td></td>
<td>36: 34: 22.7: 43: 77</td>
<td></td>
</tr>
</tbody>
</table>

文献


— 6 —