Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Gray Projection

Milad Soltany 1+, Saeid Toosi Zadeh 1 and Hamid-Reza Pourreza 2

1 Artificial Intelligence Dept, Islamic Azad University Mashhad Branch, Mashhad, Iran
2 Computer Engineering Dept, Ferdowsi University of Mashhad, Mashhad, Iran

Abstract. A fast pupil-positioning algorithm for real-time eye tracking is proposed in this paper. It is significant accurately locate the pupil position in an eye tracking system. Commonly used method is combining edge detection algorithms and ellipse fitting. Edge detection algorithms are used to detect edges of the pupil while the ellipse fitting is supposed to find the optimum ellipse that finely fits the pupil and the centre of the ellipse is regarded as the location of the pupil. This algorithm is acceptable except that the definition of the pupil edge has a great influence on its efficiency and ellipse fitting is a time consuming method. This paper focuses on accuracy of the primary algorithm and some improvement on it and uses circular Hough transform to detect pupil area. Firstly, mainly localize the pupil position by gray projection. Secondly, fit a circle to pupil using circular Hough transform.

Keywords: Eye tracking, Pupil positioning, Edge detection, Gray projection, Hough transform

1. Introduction

Eye tracking is an extensive research area in visual information system and computer vision that being used in the human-computer interface device [1]. There are two kinds of systems: head mounted eye-tracking system [2] and remote (non-contact) eye-tracking system. The non-contact eye-tracking system is preferred because it tends to be more comfortable and convenient. Unfortunately, there are still many unresolved problems preventing the use of non-contact eye-tracking systems in the actual applications such as computation complexity. Normally, eye tracking is performed on two dimensions to measure the horizontal and vertical motions of the eye. Eye (2D) position can be determined from the pupil centre coordinates, which can be computed using gray horizontal and vertical projection [3]. The most popular systems to date rely on infrared (IR) LEDs in order to extract the pertinent features needed for eye detection and tracking [3-4]. In addition to the high cost of specialized hardware, IR LEDs can raise a safety concern surrounding prolonged exposure. This can lead to an unwillingness o use an IR based system.

In the automated analysis of eye images for eye-tracking systems, detecting simple shapes such as circles arises [5]. In many cases, an edge detector can be used to obtain image points or image pixels that are on the desired curve in the image space [2,6]. Due to imperfections in either the image data or the edge detector, however, there may be missing points or pixels on the desired curves as well as spatial deviations between the ideal circle and the noisy edge points as they are obtained from the edge detector. For these reasons, it is often significant to group the extracted edge features to an appropriate set of circles. The purpose of the Hough transform is to address this problem by making it possible to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects [4,6].

Color space conversions often perform for eye detection. RGB color space [7] and HSV color space [5,8] and YCbCr color space [9-10] are the most famous color spaces.

* Corresponding author. Tel.: +98 915 3287139.
E-mail address: milad.soltany@mshdiau.ac.ir
In this paper, we propose a new method based on gray projection and circular Hough transform that doesn't use IR light and doesn't need any color space conversion. The method is used to measure horizontal and vertical pupil position in real time tracking. We attempt to minimize the complexity problem of the original algorithm while maintaining the same accuracy, by adding circular Hough transform.

The paper is organized as follows. The next presents used methods. Section 3 describes our new algorithm. Section 4 summarizes our findings and the future work.

2. Used Methods

2.1. Gray Projection

The gray projection algorithm, bases on the principle of statistics which accumulates each pixel by row or column in gray scales. The process is described as follows [3,9,11]. Given a $M \times N$ gray image $I(i,j)$, which denotes the gray scale of the pixel with the coordinates $(i,j)$, the horizontal and vertical gray projection can be defined as (1).

$$
H(i) = \frac{\sum_{j=0}^{N-1} I(i,j)}{N} \quad V(j) = \frac{\sum_{i=0}^{M-1} I(i,j)}{M}
$$

Curves have been drawn in Fig.1 (b) and Fig.1 (c) which indicates the statistics characteristics of the gray scales of horizontal and vertical pixels of Fig.1 (a).

2.2. Circular Hough Transform (CHT)

The Hough transform can be described as a transformation of a point in the x-y plane to the parameter space [12]. The parameter space is defined according to the shape of the object of interest.

A straight line passing through the points $(x_1,y_1)$ and $(x_2,y_2)$ can be described in the x-y plan by:

$$
y = ax + b
$$

This is the equation for a straight line in the Cartesian coordinate system, where $a$ and $b$ represent the parameters of the line. The Hough transform for lines does not using this representation of lines, since lines perpendicular to the x-axis will have an $a$-value of infinity. This will force the parameter space $a$, $b$ to have infinite size. Instead a line is represented by its normal, which can be represented by an angel $\theta$ and a length $\rho$ .

$$
\rho = x.\cos(\theta) + y.\sin(\theta)
$$

The parameter space can now be spanned by $\theta$ and $\rho$, where $\theta$ will have a finite size, depending on the resolution used for $\theta$. The distance to the line $\rho$ will have a maximum size of two times the diagonal length of the image. The circle is actually simpler to represent in parameter space, compared to the line, since the parameters of the circle can be directly transfer to the parameter space [13]. The equation of a circle is:

$$
r^2 = (x-a)^2 + (y-b)^2
$$

As it can be seen the circle got three parameters, $r$, $a$ and $b$. Where $a$ and $b$ are the center of the circle in the x and y direction respectively and where $r$ is the radius. The parametric representation of the circle is:

$$
x = a + r.\cos(\theta) \quad y = b + r.\sin(\theta)
$$

Thus, the parameter space for a circle will belong to $R^3$, whereas the line only belonged to $R^2$. As the dimension of the parameter space $R$ increases so the complexities of the Hough transform increase. In order to simplify the parametric representation of the circle, the radius can be held as a constant or limited to number of known radii.

To finding circles in an image using CHT, First we find all edges in the image. At each edge point, we draw a circle. This circle is drawn in the parameter space, such that our x axis is the $a$-value and the y axis is
the $b$-value while the $z$ axis is the radii. At the coordinates, which belong to the perimeter of the drawn circle, the value in the accumulator matrix will be increased. The accumulator will now contain numbers corresponding to the number of circles passing through the individual coordinates. Thus the highest numbers (selected in an intelligent way, in relation to the radius) correspond to the centre of the circles in the image. The algorithm for Circular Hough Transformation can be summarized to:

- Find edges. Then for each edge point //HOUGH BEGIN
- Draw a circle with center in the edge point with radius $r$ and increment all coordinates that the perimeter of the circle passes through in the accumulator
- Find one or several maxima in the accumulator //HOUGH END
- Map the found parameters $(r,a,b)$ corresponding to the maxima back to the original image

3. Description of Approach

In order to avoid harming the eyes, reduce costs and improve accuracy, we explore the use of natural light to detect the eyes of human. Using RGB color space is another simplifying of our approach that removes complexity of color space conversion.

3.1. Pupil Segmentation

Pupil part is obviously different from the rest in the gray scale eye image. One typical specialty is the darkness of the pupil, which results in low gray scales in the gray image. Fig.1 (a) is a gray scale eye image, while Fig.1 (d) shows the corresponding gray histogram of it. A wave crest can clearly be seen in Fig.1 (d) (marked by a red rectangle), which mostly distinguishes the pupil area from the rest of the eye image.

Therefore, the part of pupil could be separated simply by classifying the pixels of the eye image with the gray threshold selected according to Fig.1 (d). The pupil part is found through image segmentation, as Fig.3 (b) shows.

Fig. 1: (a) A gray scale eye image (b) Horizontal gray projection (c) Vertical gray projection (d) the corresponding gray histogram of eye image

Fig. 2: (a) Original eye image and (b) Binary image with thresholding (c) Enhanced pupil binary image and (d) Approximate center point of the pupil

Fig.3 (b) show segmentation isn't enough for finding pupil area. By calculating the maximum value of curves in the Fig.1, exact boundary box of pupil area can be defined as (6).
3.2. Primary Pupil Positioning Using Gray Projection

After eye image segmentation, gray projection curves can be drawn e.g. Fig.1 which indicates the statistic characteristics of the gray scales of vertical and horizontal pixels. It shows that the value of the projection function has remarkably changed when passing through the pupil area. As a result, the coordinates of pupil can be determined approximately by choosing the center of rectangle. The green cross mark in Fig.3 (d) is the primary positioned center point of pupil.

3.3. Find Exact Pupil Position

The accuracy of pupil positioning is mostly decided by the edge of pupil. It is generally reckoned that the point of which the pixel has the large gray gradient is the edge point of the pupil. One of the edge detection operators is Canny operator that applies two thresholds to the gradient: a high threshold for low edge sensitivity and a low threshold for high edge sensitivity.

After detecting edges, it is time to use circular Hough transform to find the pupil exactly. Fig.5 (a) shows the result of executing CHT and show executing CHT directly on image produces a lot of circles that all of them is wrong except one circle that fit on pupil. One solution for this problem is to limit the area of image that CHT implemented on it. We limit area using the result boundary box in section A and Fig.5 (b) shows the result.

Euclidean distance can be used to calculate distance between the center of circles and approximate center of pupil that can be expressed as (7).

\[
dist(ac, cc) = \sqrt{(x_{ac} - x_{cc})^2 + (y_{ac} - y_{cc})^2}
\]  

(7)

Where \(ac(x_{ac}, y_{ac})\) is coordinate of the approximate pupil center and \(cc(x_{cc}, y_{cc})\) is coordinate of the circle center. After calculate distance, nearest circle found by CHT is fitted to pupil and Fig.5 (d) show the results.
4. Performance Analysis

We used a Genius e-Messenger 310 camera to capture user images and the format of input image is 640×480 RGB color. The algorithm is implemented on a PC with AMD Turion 2.0 GHz CPU and 2 GB of RAM, with a Microsoft Windows XP operating system and has been developed and tested in Matlab R2009b.

Experiments on different eye images without using infrared light are executed to testify the improvement of ellipse fitting of pupil. Fig. 7 shows four eye pictures from different subjects under various illuminations.

![Fig. 4: Results of circle fitting of different eye images with CHT](image)

The main emphasis of this work was on enhancing the robustness and complexity of the algorithm for detecting the pupil boundary. With complexity reduction, it is clear that algorithm speeds up and TABLE I shows results.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Original algorithm [3]</th>
<th>Our proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time consumption</td>
<td>9.5 ~ 11 ms</td>
<td>6.5 ~ 8 ms</td>
</tr>
<tr>
<td>Code complexity</td>
<td>O(n^2)</td>
<td>O(n^2)</td>
</tr>
</tbody>
</table>

In order to evaluate the performance of the improved algorithm, we use Euclidean distance between the actual pupil center and calculated center with algorithms and define:

\[ MP = \frac{\sum_{i=1}^{n} \text{EucDst}(pc_i, cc_i)}{n} \quad (8) \]

Where \( MP \) is the average misplacement of calculated the pupil center, \( n \) is the number of image, \( \{ pc_i, cc_i | i=1,…,n \} \) is the actual pupil center and calculated pupil center by algorithms respectively and \( \text{EucDst} \) is Euclidean distance in (8). Each algorithm that has less \( MP \) is better than another. Table 1 shows average misplacement of 20 test images. It can be seen that our proposed algorithm estimates center of pupil better than original algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Original algorithm [3]</th>
<th>Our proposed algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>( MP )</td>
<td>0.6 ~ 2.6</td>
<td>0.3 ~ 1.7</td>
</tr>
</tbody>
</table>

5. Conclusion

Some approaches to pupil positioning have been discussed in this paper. We propose non-IR based algorithm and use RGB color space. First, the boundary box and approximate center of pupil in the eye image are confirmed by steps of image segmentation and gray projection. Second, Canny operator is used to detect the edge of the pupil. Finally, the pupil is localized via circular Hough transform. The next step would be to implement the algorithm in C and optimize the functions involved.

Experiments indicate that the improved algorithm is more accurate and less time-consuming.
6. References


