

A Novel Accurate Algorithm to Ellipse Fitting for Iris Boundary Using Most Iris Edges

Mohammad Reza Mohammadi¹, Abolghasem Raie²

1. Department of Electrical Engineering, Amirkabir University of Technology, Iran. mrmohammadi@aut.ac.ir
2. Department of Electrical Engineering, Amirkabir University of Technology, Iran. raie@aut.ac.ir

Abstract: Ellipse fitting for iris boundary is an important step in eye gaze estimation and iris recognition applications. In most of the proposed algorithms only the edge image is used for ellipse fitting. In these algorithms, the undesired edges may lead to erroneous iris ellipse. In this paper, we propose a novel algorithm to ellipse fitting for iris boundary which uses gray and edge images simultaneously. In this algorithm, edges of eye region are computed and refined by some proposed constraints. Then, remainder segments merge together by an optimization process and the best ellipse is fitted for the merged segments. In this algorithm, most edges of iris boundary are used for ellipse fitting and hence, its accuracy is high. The performance of the method has been evaluated on real images. *J Am Sci* 2012;8(6):758-763]. (ISSN: 1545-1003). <http://www.americanscience.org>. 94

Keywords: Ellipse fitting, Split points, Merge of segments.

1. Introduction

In applications such as eye gaze estimation [1-5] and iris recognition [6-9] an important step is fitting a shape for iris boundary. In some papers, it is assumed that contour of iris is a circle (such as [3]). But, as shown in [1], contour of iris in a 2D image is an ellipse.

In [1,2,6] edges of the eye region image are computed by simple algorithms and then ellipse fitting algorithms are proposed to apply on it. In [1], with respect to eye shape and eyelid effects, it is assumed that two longest vertical segments belong to iris boundary. Then, an ellipse is fitted for the pixels on these segments by the least square method. Not using horizontal pixels lead to reduced accuracy of this algorithm. Also, in some images, it is possible that an erroneous segment is selected for iris boundary and the error of the ellipse fitting becomes very large. An improvement of Randomized Hough Transform (RHT) is proposed in [2] to ellipse fitting for iris boundary and in [6] RANSAC algorithm is used for it.

A common problem in the presented algorithms is using only edge image to fitting the desired ellipse. In some images, because of eyelids and other edges, the best fitted ellipse to the edge image is not identical to the iris ellipse. To overcome this problem, it is required to use gray (or color) image and edge image simultaneously. In this paper, we propose a new algorithm to ellipse fitting for iris boundary which utilizes gray and edge images. In this algorithm, we compute edges of eye region and remove "split points" (defined in section 3) from it. In addition, we remove some segments, with respect to iris center, which cannot belong to the iris boundary. Final step is merging the remaining segments and fitting the best ellipse for them.

The remainder of this paper is structured as follows. In Section 2, we propose a novel algorithm to compute and refine edges of eye region which eliminates a lot of undesired ones. In Section 3, an optimization algorithm is introduced to merge the remainder segments. Experimental results are presented in Section 4. Finally, Section 5 concludes this paper.

2. Edge Computation and Refinement

In this section, we present a novel algorithm for edge computation and refinement formed from three steps. The first step is to compute edges. Elimination of split points and undesired segments are the next steps discussed in the remainder of this section.

2.1. Edge Computation of Eye Region

Because of the existence of a lot of undesired edges in the eye region, most of them related to eyelids and eyelashes, it is required that the eye region edges be computed by high accuracy and be refined. To compute the edges, we use Canny operator in a 2-level pyramid. In other words, by Canny operator, edges on low-resolution image are computed. Then, edges on high-resolution are computed and edges which do not have equivalent pixel in low-resolution image are removed.

By this method, a lot of small edges, most of them related to eyelashes, are removed. Figure 1 shows an example of edge computation by this method.

Some pixels computed as edge in eye region image do not belong to iris boundary (undesired pixels). To eliminate a lot of them, we propose new algorithms. Undesired pixels are located on segments which all or parts of them are undesired. Thus, for

any segment, we compute pixels which are the location of the intersection of two (or more) different segments. After this, all pixels in a segment are undesired or not. Hence, we eliminate all pixels of the segments which their probability to be located on iris boundary is lower than the threshold. These operations will be discussed in the next parts of this section.

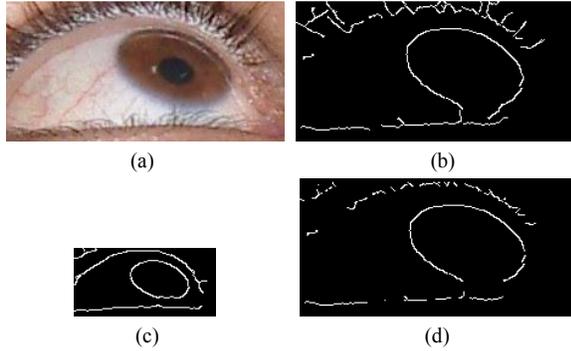


Figure 1. Edge computation by Canny operator in a 2-level pyramid, (a) main image, (b) edges of high-resolution image, (c) edges of low-resolution image, and (d) result of combination of two levels

2. 2. Elimination of Split Points

In the edge image of eye region, there are two types of split points (Figure 2). Type 1 is the location of the intersection of two (or more) different segments and type 2 is the location in which the end parts of two different segments are identical.

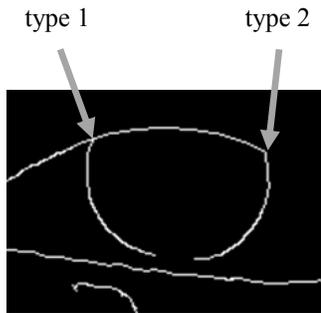


Figure 2. Two different types of the split points in an edge image

To compute the location of type 1 split points, we make the edge image thin by morphological thinning operator and convolve the result by a 3x3 ones filter. Pixels, in the resulted image, which have values greater than 2 are the locations of split points of type 1.

The main parameter to localize split points of type 2 is the change of differential vector. The change of differential vector in an ellipse cannot be greater than a threshold. So, we use the change of

differential vector at the pixel and the average differential vector for some pixels before and after the pixel to determine split points of type 2.

The first parameter consists of differences in differential vectors for current and next pixels. In discrete points, differential vector is obtained from Equation (1):

$$d\vec{l}_i = dx_i \hat{i} + dy_i \hat{j} \tag{1}$$

where dx_i and dy_i can be calculated from Equations (2) and (3).

$$dx_i = x_i - x_{i-1} \tag{2}$$

$$dy_i = y_i - y_{i-1} \tag{3}$$

To quantify the changes of differential vectors, we use normalized dot product as Equation (4):

$$d_i = \frac{dx_{i+1}dx_i + dy_{i+1}dy_i}{\sqrt{(dx_{i+1}^2 + dy_{i+1}^2)(dx_i^2 + dy_i^2)}} \tag{4}$$

If the number of points for an ellipse is large, d_i should be approximately 1 for all points. But, if the ellipse consists of only 20 pixels and its aspect ratio is smaller than 2 (equivalent to $\theta = 60$ degrees), the values of d_i change theoretically between 0.8176 and 0.9875. Thus, we set $T_d = 0.8$ as threshold of d .

Differential vector at a point has high sensitivity to noise and other errors in the edge detection phase. Therefore, we use the average differential vector as the second parameter. To obtain a good estimation of the average differential vector, we fit a line in the sense of MSE for k points before the current pixel. By some calculations, Equation (5) results the slope of the fitted line:

$$m_1 = \frac{\sum_{l=1}^k lx_{1l} - \frac{k+1}{2} \sum_{l=1}^k x_{1l}}{\sum_{l=1}^k ly_{1l} - \frac{k+1}{2} \sum_{l=1}^k y_{1l}} \tag{5}$$

where x_{1l} and y_{1l} are coordinates of the previous points. In the same manner, Equation (6) is obtained for k points after the current pixel:

$$m_2 = \frac{\sum_{l=1}^k lx_{2l} - \frac{k+1}{2} \sum_{l=1}^k x_{2l}}{\sum_{l=1}^k ly_{2l} - \frac{k+1}{2} \sum_{l=1}^k y_{2l}} \quad (6)$$

The amount of difference between these two slopes is computed by normalized dot product as in Equation (7):

$$m = \frac{|1 + m_1 m_2|}{\sqrt{(1 + m_1^2)(1 + m_2^2)}} \quad (7)$$

If m is close to 1, it means that the average slope changes slowly. Otherwise, the average slope changes rapidly and means that this pixel can be a split point of type 2. Based on the experiments, if all points of the ellipse are greater than 50, selecting 4% of them is sufficient for k . With this choice, for ellipses with an aspect ratio smaller than 2, m changes between 0.8800 and 0.9921. So, we select $T_m = 0.85$ as threshold of m .

With this description, summary of the algorithm is as follows:

For all pixels of any segment, value of d is computed from Equation (4) and compared with T_d .

If $d < T_d$, then, value of m is computed from Equation (7) and compared with T_m .

If $m < T_m$, then, a split point of type 2 has detected.

Figure 3 shows an example of applying the proposed algorithm for elimination of the split points.

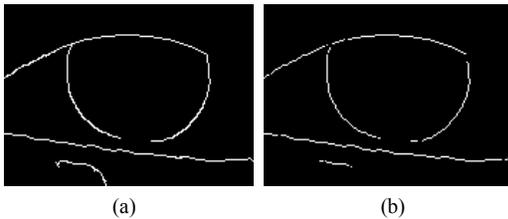


Figure 3. Elimination of the split points by the proposed algorithm, (a) an edge image of eye region, and (b) elimination of split points

2.3. Elimination of Undesired Segments

Segments that belong to iris boundary have curvature towards the iris center. Therefore, to eliminate undesired segments, we estimate iris center, then compute the amount of curvature for any segments and compare it by a threshold value.

In this algorithm, accuracy of the iris center localization is not essential. Thus, we use a simple and fast algorithm for it. In the eye region, the darkest parts belong to pupil and iris regions; therefore, we convolve the image by a large Gaussian filter and compute the location of minimum value in

it as iris center. Two samples of iris center localization by this method are shown in Figure 4(a).

For a pixel located on perimeter of an ellipse, the slope is approximately perpendicular to the line that connects center of ellipse to it. Moreover, concavity in the pixels of an ellipse is approximately towards the center of the ellipse. So, we fit two parametric curves of order 3 for x and y coordinates of the segment. Then, we compute first and second order derivatives of the fitted curve and choose them to call dx_t, dy_t, d^2x_t and d^2y_t , respectively: where $t = 1 : n$ while n is the number of pixels of the segment.

To quantify the first and second features (perpendicularity of the slope to and parallelism of the concavity by the line that connects center of ellipse to the pixel) we compute normalized dot product for each pixel as Equations (8) and (9):

$$s_{t1} = (x_t - x_i)dx_t + (y_t - y_i)dy_t \quad (8)$$

$$s_{t2} = (x_t - x_i)d^2x_t + (y_t - y_i)d^2y_t \quad (9)$$

Final parameter is calculated from weighted average of s_1 and s_2 as in Equation (10):

$$s = \frac{1}{n} \sum_{t=1}^n k_1 s_{t1} + k_2 s_{t2} \quad (10)$$

According to the performed experiments, we choose $k_1 = 5$ and $k_2 = -1$ and set threshold value to -3 . Figure 4(b) shows the results of applying this algorithm on the images of Figure 4(a).

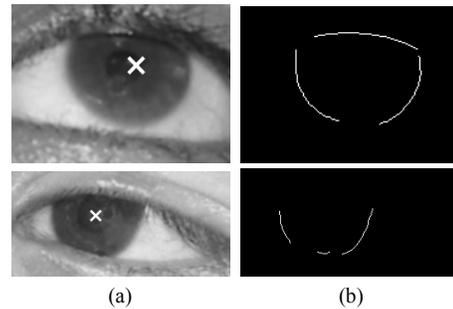


Figure 4. Elimination of the undesired segments, (a) iris center localization by a simple filtering, and (b) remainder segments

3. Merging of the Remainder Segments

In previous section, we computed and refined the edges of eye region image. The output of these algorithms is several segments that only some of them belong to iris boundary. In this section, we propose an algorithm to merge the segments that belong to iris boundary. In [1], two longest vertical edges are selected as iris boundary. The performance of this method, for heads without rotation, has been

reported to be appropriate. Sensitivity to head rotation and not using horizontal edges are the main limitations of this method.

In this paper, for maximum using of iris edges and eliminating limitation of head rotation, we define a fitness function whose optimization results the segments that should be merged. In other words, the fitness function is defined according to parameters of each segment. Then, only segments that improve the fitness function merge together. We use four features in the fitness function as discussed in next parts.

3.1 Average Distance from the Fitted Ellipse

The first feature to merge two segments is the average distance of the points in the segments from their fitted ellipse. This feature expresses elliptical measure of the points in the segment. It is important that both segments be close to the fitted ellipse. So, we propose Equation (11) as f_1 :

$$f_1 = (diff_1 + diff_2) \times \sqrt{\frac{\max(diff_1, diff_2)}{\min(diff_1, diff_2)}} \quad (11)$$

Where $diff_1$ and $diff_2$ are average distances of the two segments from the fitted ellipse. For two segments that are located on an ellipse, the value of f_1 will be 0. Fitted ellipses to four combinations of two segments are shown in Figure 5. For these combinations, values 0.1942, 0.1723, 1.2855 and 1.6020 are computed respectively as f_1 which expresses high performance of this feature.

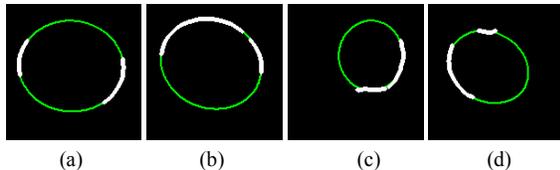


Figure 5. Values of f_1 for 4 examples, (a) 0.1942, (b) 0.1723, (c) 1.2855, and (d) 1.6020

3.2. Matching of the Fitted Ellipse by All Edges

The second feature is the percent of points in all segments that are close to the fitted ellipse of the segments. In other words, this feature determines how many points of all segments match with this ellipse. So, we propose Equation (12) as the second feature:

$$f_2 = \frac{\sum_{i=1}^L (diff_i < T)}{L} \quad (12)$$

where L is the number of all points on the segments and $diff_i$ is the distance of the i th point from the fitted ellipse. Also, T is a threshold level whose value is 0.5. Figure 6 shows an edge image that consists of 5 segments. Parts (b) to (f) show 5 combinations (out of 10 possible combinations) whose f_2 values are 0.3988, 0.4104, 0.0809, 0.3873 and 0.1908, respectively. The best value for f_2 is 1; so, the computed values show the high performance of this feature.

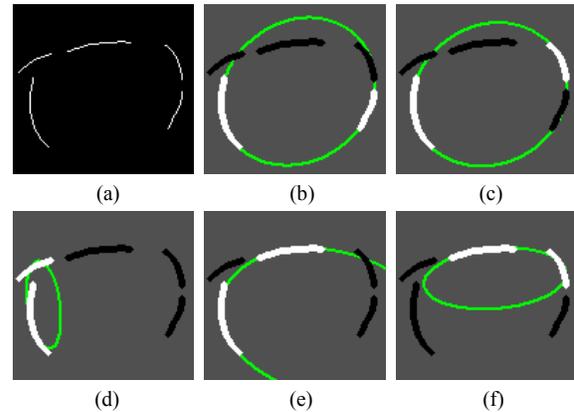


Figure 6. Matching of the fitted ellipse by all edges, (a) an edge image which consists of 5 segments, and (b) to (f) five combinations for which values 0.3988, 0.4104, 0.0809, 0.3873 and 0.1908 are obtained as f_2

f_1 and f_2 features use only the edge image to merge the segments. In many cases these features are sufficient; however, in some cases the best fitted ellipse for the edge image is not identical to the best fitted ellipse to iris boundary. Therefore, we define the third and fourth features which are based on gray image of the eye region.

3.3. Adjacency to the Eyeball Region

An important feature of the iris edges is their adjacency to the eyeball region. Moreover, eyeball is the brightest region in the eye image. So, we can extract eyeball region by a simple thresholding algorithm and give high scores to the segments whose pixels are close to this region. We propose (13) as f_3 :

$$f_3 = N_i + N_j \quad (13)$$

Where N_i and N_j are the percent of points in each segment being close to the eye region. Figure 7 shows a sample of computing N_i . Values 0.9268, 0, 0, 1 and 0 are computed as N_i for the 5 segments of this figure, respectively. These values indicate the robustness of this feature.

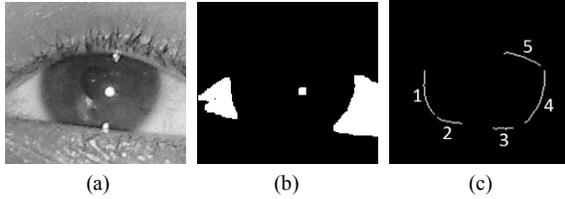


Figure 7. Adjacency to the eyeball region, (a) original image of eye region, (b) thresholded image followed by opening operator to remove small particles, and (c), five remainder segments whose values of f_3 are 0.9268, 0, 0, 1 and 0, respectively

3. 4. Difference of Iris Color by Other Segments

Another important feature for iris ellipse is color of pixels located inside and outside of it. Thus, we define the ratio of average intensity for around pixels of the fitted ellipse located inside and outside of it as f_4 . Around iris boundary, inside pixels are darker than outside ones; therefore, this feature can remove segments whose fitted ellipse is not compatible with eye properties. Equation (14) is proposed to compute f_4 :

$$f_4 = \frac{M_i}{M_o} \quad (14)$$

where M_i and M_o are the average intensities for pixels of inside and outside of the fitted ellipse. Figure 8 shows some samples to demonstrate the performance of this feature. Parts (a) and (b) are the gray image and edge image that consists of 5 segments, respectively. Parts (c) to (f) are 4 combinations (out of 10 possible combinations) for which the values 0.5158, 0.5219, 0.8641 and 0.5951 are computed as their f_4 . So, this is a good discriminative feature.

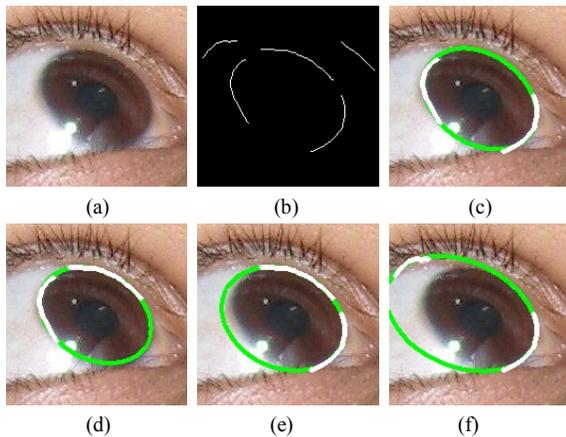


Figure 8. Performance of the f_4 for some examples, (a) original image of eye region, (b) the edge image which consists of 5 segments, and (c) to (f) four combinations for which values 0.5158, 0.5219, 0.8641 and 0.5951 are obtained as f_4

3. 5. Optimization of the Fitting Function

We define the fitting function as a weighted sum of the discussed features. So, Equation (15) is the fitting function:

$$f = Af_1 + Bf_2 + Cf_3 + Df_4 \quad (15)$$

To optimize f , we merge only the segments that increase the value of f . With respect to the importance of the features and preformed experiments, we choose values -3, 2, 1.5 and -2 for A , B , C and D , respectively. Implementation results show that the accuracy of this method is very high and it can manage the existence of many undesired edges.

Figure 10 shows different steps of the proposed algorithm to ellipse fitting for iris boundary in a sample image. In part (d) of this figure, merged segments are shown with white color and other segments with gray color. Also, the fitted ellipse, whose parameters are computed for pixels on the merged segments by algorithm of [10], is shown with black color.

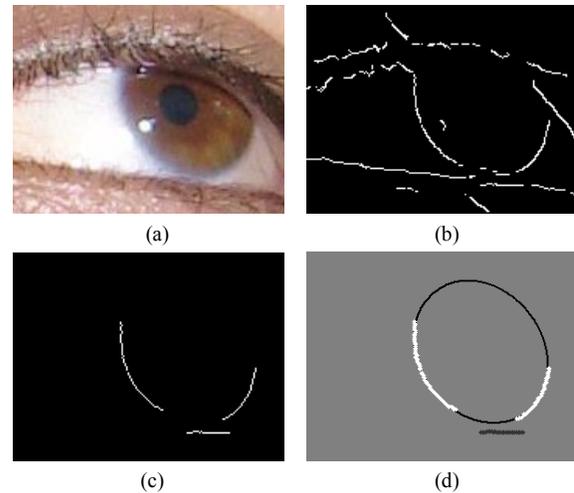


Figure 9. Different steps of the proposed algorithm for ellipse fitting to iris boundary, (a) original image of eye region, (b) edge computation, (c) edge refinement, and (d) merge of remainder segments and the final fitted ellipse

4. Experimental Results

The main motivation of this paper is to propose a new algorithm that, independent of head rotation, uses most of the existing iris edges to fit an accurate ellipse to the iris boundary. To evaluate the proposed algorithm, we get 72 images from eye regions. For any image, the number of all edges, the number of iris edges and the number of split points (consisting of pixels located on undesired segments) are counted.

For all images, from 18201 edge pixels, 1406 pixels are eliminated as split points and 2518 pixels are removed as undesired segments. Moreover, 11860 pixels identified as the iris edges. It should be noted that 2417 pixels are removed in merging phase. The error of the proposed algorithm was 112 pixels which erroneously have been considered as iris edges and 120 pixels which belong to iris boundary but are not considered. Thus, FPR and FNR of the proposed algorithm are about 1%.

5. Conclusion

The main contribution of this study was the development of an algorithm to accurately estimate the parameters of the ellipse of the iris boundary. The novel aspect of this work was proposing new constraints to split the edge image to some segments and proposing new features to merge these segments.

One of the future works is defining some new features to use in the fitness function. Moreover, to eliminate the split points we use the change of differential vector. To improve this algorithm, we can use gray image features in the future works.

Acknowledgements:

The authors would like to express their sincere thanks to Dr M. R. Eslami Khouzani for their valuable comments.

References

1. Wang J.G, Sung E, Venkateswarlu R. Estimating the eye gaze from one eye. *Comput Vis Image Underst* 2005; 98(1):83-103.
2. Huang S.C, Wu Y.L, Hung W.C, Tang C.Y. Point-of-Regard Measurement via Iris Contour with One Eye from Single Image. in *Proceedings of the 2010 IEEE International Symposium on Multimedia*, 2010.
3. Liang D.B.B, Lim Kok H, Non-intrusive eye gaze direction tracking using color segmentation and Hough Transform. in *Communications and Information Technologies, 2007. ISCIT '07. International Symposium on*, 2007;602-607.
4. Kohlbecher S, Bardinst S, Bartl K, Schneider E, Poitschke T, Ablassmeier M. Calibration-free eye tracking by reconstruction of the pupil ellipse in 3D space. in *Proceedings of the 2008 symposium on Eye tracking research & applications*, Savannah, Georgia, 2008.
5. M. R. Mohammadi, and A. Raie, Robust Pose-Invariant Eye Gaze Estimation Using Geometrical Features of Iris and Pupil Images, the 20th Iranian Conference on Electrical Engineering, ICEE2012, 2012.
6. D. Zhang, A. Jain, and X. Li, Modeling Intra-class Variation for Nonideal Iris Recognition, *Advances in Biometrics, Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2005; 419-427.
7. Bonney B, Ives R, Etter D, Yingzi D. Iris pattern extraction using bit planes and standard deviations. in *Signals, Systems and Computers, Conference Record of the Thirty-Eighth Asilomar Conference on*, 2004;582-586.
8. Daugman J. New Methods in Iris Recognition. *Systems, Man, and Cybernetics, Part B: Cybernetics*, IEEE Transactions on, 2007;37(5):1167-1175.
9. Pillai J, Patel V, Chellappa R, Ratha N. Secure and Robust Iris Recognition using Random Projections and Sparse Representations. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 2011;(99):1-17.
10. Fitzgibbon A, Pilu M, Fisher R.B. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, 1999; 21(5):476-480.

5/15/2012