An Efficient Iris and Eye Corners Extraction Method

Nesli Erdogmus[1], Jean-Luc Dugelay[1],

[1] Eurecom, Multimedia Communications Department[*]
2229 Routes des Crêtes, 06904 Sophia Antipolis, France
{nesli.erdogmus, jean-luc.dugelay}@eurecom.fr

Abstract. Eye features are one of the most important clues for many computer vision applications. In this paper, an efficient method to automatically extract eye features is presented. The extraction is highly based on the usage of the common knowledge about face and eye structure. With the assumption of frontal face images, firstly coarse eye regions are extracted by removing skin pixels in the upper part of the face. Then, iris circle position and radius are detected by using Hough transform in a coarse-to-fine fashion. In the final step, edges created by upper and lower eyelids are detected and polynomials are fitted to those edges so that their intersection points are labeled as eye corners. The algorithm is experimented on the Bosphorus database and the obtained results demonstrate that it can locate eye features very accurately. The strength of the proposed method stems from its reproducibility due to the utilization of simple and efficient image processing methods while achieving remarkable results without any need of training.

Keywords: Eye features extraction, iris, eye corners and eyelids

1   Introduction

Facial features have a crucial role in many computer vision applications such as face normalization, facial expression recognition or model-based human face coding. For this reason, automation of their extraction has a wide range of usage. Among those features, eyes have the highest importance with their higher prominence and stability compared to other facial features. In [1], it is proven that the eyes can improve the recognition performance as compared to the nose and mouth. The eyes features include iris center (or pupil center) and radius, eyelid contours and eye corners which are located at the intersection of the upper and lower eyelids.

In most cases, firstly the eye region is extracted. Many different methods have been proposed for this task, such as extracting contrasted components by morphological operations [4], using eye filters to detect eye candidates [5] or by

---

[*] http://image.eurecom.fr

projecting the facial edge map vertically and horizontally, where the maximum points of the projection curves are associated to the eye positions [6].

For iris, there are numerous approaches where the center and the radius are searched separately [2] or together [3]. In [2], the physiological property of the pupil is used to detect the center. Due to the pupil's response to the light, it is the brightest region in H channel of the HSV color space. After the detection of the center, iris radius is estimated so that the mean gray level of the pixels in the circle is the lowest. On the other hand in [3] circular Hough transform is used to detect the iris border where both center and radius are estimated simultaneously. In some approaches, the iris radius is supposed to be known [8] or limited to a set of expected values [7].

In order to locate eye corners, one general approach is projection functions [9]. The weakness of Integral Projection Function (IPF) to reflect well the variation in the image Variance Projection Function (VPF) is proposed [10]. Later this approach is diversified as General Projection Function (GPF) and Hybrid Projection Function (HBF) which combine IPF and VPF and Weighted Variance Projection function (WVPF) [11] in which pixels are assigned weights according to their Harris corner response. Utilization of deformable templates [7] is another common approach to detect eye corner positions which often requires a good initialization in order to avoid incorrect results. Additionally, in [2] eye-corner filter using Gabor feature space is proposed for eye-corner detection and in [12], two semantic features for eye corners are introduced which are further fused by logistic regression classifier to determine their accurate locations.

Lastly, for eyelids the proposed methods can be classified under two groups: using deformable contour models [13], curve fitting [7, 14]. As mentioned before, for deformable contour models initialization is crucial. Additionally, the energy term should be formulated carefully to reach an accurate result. In curve fitting approach, usually the eyelid contours are extracted after the detection of the eye corners. In [7, 14], parabolic sections with parameters controlling its curvature, position and rotation is fitted to a set of points including the corner points and the detected edges in the eye region whereas in [15] edges are replaced by four control points where iris border and the eyelids intersect.

In this paper, the facial region in the image is assumed to be known and the eye region is taken to be the non-skin region in the upper half of the facial image with the assumption of frontal face with the nose being vertical. Firstly, a coarse localization of the irises is performed in the estimated eye region by circle detection using Hough transform. The detected circles are subjected to elimination with the help of a priori knowledge about relative size and position of irises. Afterwards, the color images of the eye regions (window around the coarsely detected iris centers) are further processed to refine the iris radius and location. Finally, the cropped eye images are segmented into three color regions and contrary to previous works, the eyelid contours are estimated first to obtain the eye corners on their intersection points.

The rest of this paper is organized as follows: In section 2, the method for detection of the eye regions by extracting the non-skin part of the face is presented in detail. The coarse estimation of the iris centers and radiuses and afterwards the refining of these results are explained in section 3. Section 4 is on eye corners detection method. Finally, section 5 is where the conducted tests and their results are represented, followed by the conclusions in section 6.

2   Eye Region Extraction

The eye region in the facial image is extracted under the assumptions that the face is frontal with the line connecting the eye centers close to horizontal. Hence, the upper half of the face is taken to be analyzed. The non-skin region is found by removing the pixels with the most frequent (C_b, C_r) values present in the image, using YC_bC_r space. For this purpose, firstly the histogram is calculated for distribution analysis. Even though the face image is cropped into its upper half where the eyes are located, still the skin pixels constitute the majority. Taking the histogram into account, a threshold is set according to the maximum count and the image size. Afterwards, the pixels with higher value than this threshold is eliminated as skin pixels. Lastly, the small islands in the obtained binary mask are removed. In Fig. 1, an example set of images is given to demonstrate the process (in which the forehead is not shown to have better view of the eye region).
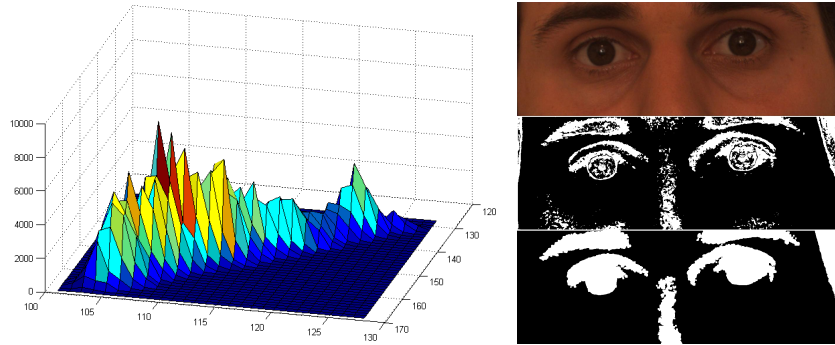

Fig. 1. A (C_b, C_r) histogram and the resulting mask after thresholding. As you can see, the non-skin region which includes eyes is clearly separated from the rest.

Since the algorithm proposed in this paper is stepwise, the iris detection results affect the rest. Hence, this part is added to the system as a supportive module to improve the iris extraction by removing other possible circular edges as much as possible. The improvement due to this addition can be seen in "Tests and Results" section.

3   Iris Extraction

After obtaining the eye regions, firstly edge maps are constructed by Canny edge detector. The drawback of this edge detection method is that it requires a good adjustment of the threshold. In order to overcome this issue, we propose to use the edge detector iteratively, by tuning the threshold parameter until a descriptive edge map is obtained. Afterwards, Hough transform is applied to the edge map to detect circles. For each detected circle, an overlapping score is calculated by the ratio of the

detected portion of the circle to the whole circle parameter. Here, we define the "descriptiveness" of an edge map by the number of the edge pixels in the image and the number of circles that can be detected using these edges. For circle detection, minimal and maximal radius values are defined to speed up the process.

Subsequently, the detected circles (iris candidates) are grouped into two classes according to their position: right side and left side. Then, for all possible pairs of right and left circles, those criteria are applied:

- Vertical distance of the centers
- Horizontal distance of the centers
- Difference between radiuses

Among the compatible pairs, the one with maximum total overlapping score is chosen to be the two irises. In Fig. 2, the procedure to roughly obtain the iris positions and dimensions is depicted with examples.
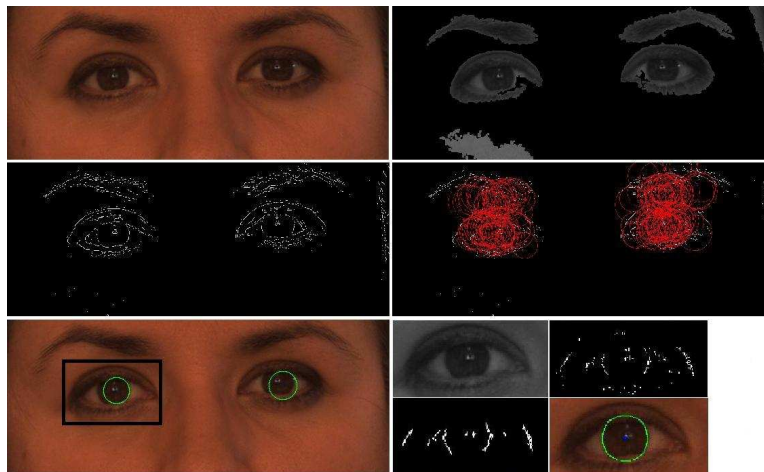

Fig. 2. From left to right, top to bottom: a. Input image b. Masked image after skin region removal c. Detected edges d. Detected circles e. New eye region window f. Refining of the iris position and radius after detecting best circle to detect vertical edges

Once the approximate positions of the irises are obtained, rectangular windows centered at the detected iris centers are extracted and analyzed separately. Firstly, an averaging filter is applied with a rectangular kernel, where the noise and horizontal edges are suppressed and vertical edges are preserved to some point. Then, the vertical edges are detected by using the Sobel operator. As also explained in [7], the upper and the lower parts of the iris border are mostly occluded by eyelids. This leads to incorrect hints for circle detection (Fig. 3). Therefore, only the vertical edges are detected. Then, the obtained edge map is cleaned with the help of morphological operations where only the connected components which are larger than a threshold, are preserved. Since edges detected by the Sobel operator are often broken, vertical dilation is applied before the removal of small islands.
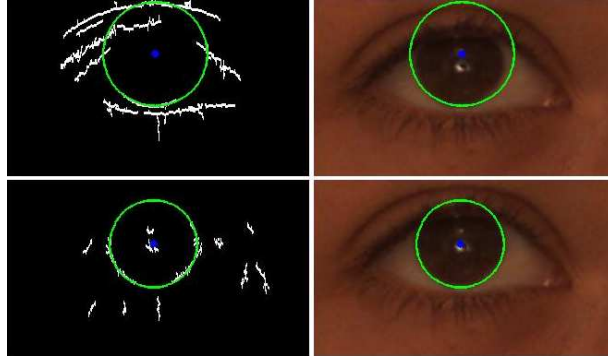
Fig. 3. The positive effect of using vertical edges only can be observed when the two detected iris circles are compared

Using this edge image, similar to the previous approach, circles are again detected by using the Hough transform method. The circle with the maximum score provides us the center and the radius of the iris.

4   Eye Corners Extraction

For this part, the eye images are further cropped since now the accurate iris centers and radiuses are known. In this approach, firstly the eyelids contours are aimed to be detected which can be used to determine the eye corners. For this purpose, the edges created by the eyelids are searched for. The edge detection is done in two ways:

- On the color segmented image
- On the grayscale image

The details are given in the following sections:

4.1   Eyelid detection on the color segmented eye images

Firstly, the color eye image is segmented into 3 regions: dark regions like iris and eye lashes, skin regions and sclera (white part of the eye ball). In this segmentation, at the beginning the input eye image is coarsely represented using 10 bins. For this coarse representation, spatial information from a Histogram based windowing process is used [16]. Next, k-means is used to cluster the coarse image data. The cluster centroid locations are initialized with the mean value of the 70 manually collected colors for each region.

After clustering, the resulting segmented image is convolved with horizontal and vertical Sobel operators to detect the corresponding edges and for each edge its angle is calculated. In view of the fact that eyelids are mostly closer to horizontal, only the edges with less than 45 degrees are taken into account. Additionally, similar to the

processing in section 3 for iris extraction, horizontal dilation is applied to connect broken edges and then small sections are removed.

4.2   Eyelid detection on the grayscale eye images

In a similar manner to the segmented ones, in this part grayscale eye images are processed to detect the eyelids. Horizontal edges are detected again using Sobel operators. Since the edges are not as well-defined, small parts of the iris border are also detected as horizontal. In order to solve this, the edges detected in the close neighborhood of the previously detected iris contour are removed. Lastly, morphological thinning operation is applied on the resulting edge map.

4.3   Final edge map and eye corners detection

After the two detected edge maps are superimposed, the following method is applied to remove outliers:

Observing that the eye corners are mostly located lower than the iris center, two lines are created, which are imagined to be approximately connecting the iris center and the corners. The slope of both lines is empirically chosen to be 1/3. Afterwards, only the closest edges that are below and above these lines are labeled as upper and lower eyelids. This method is illustrated in Fig. 4.
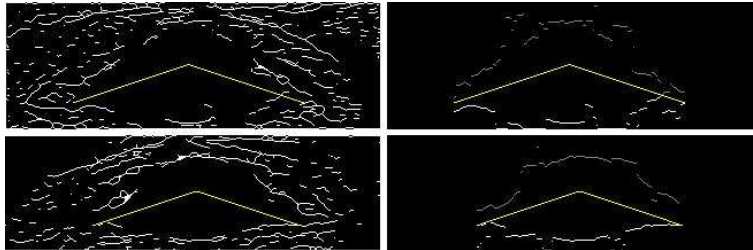

Fig. 4. Two example edge maps before and after the method is applied

In the final step, 2nd and 3rd degree polynomials are fitted for lower and upper eyelids edges respectively, in a least squares sense. The fitting is repeated once more with only the edges close to the first estimation, to further remove the outliers that still exist. The inner (near the nose bridge) and outer eye corners are determined as the intersection points of the two fitted polynomial. In Fig. 5, a set of sample images is given to illustrate each step of this section.
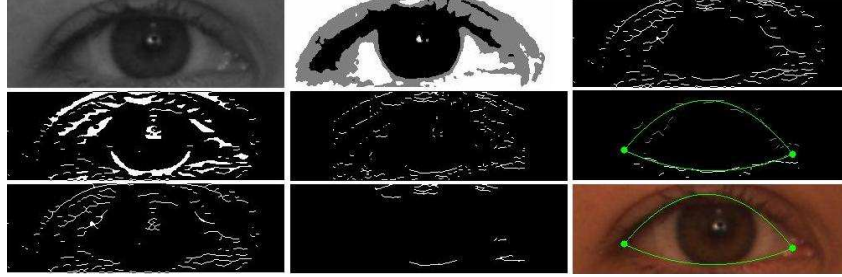
Fig. 5. Each column from left to right: a. Edge detection using grayscale image and removal of the edges close to the iris contour b. Detection of edges with less than 45° using segmented image and removal of small sections c. Fusing the two edge maps and curve fitting after eliminating edges that are not related to eyelids.

5   Tests and Results

The method proposed in this paper is tested on the Bosphorus Database [17] which is actually a database of 3D faces but also supplies 2D high resolution color images which are cropped for face regions. The presented eye features extraction approach is applied to the neutral and frontal images of 105 subjects in the database. The image sizes are not fixed and they change between (936-1404) $\times$ (1218-1740) pixels. The computational time in Pentium(R) Dual-Core CPU 2.49GHz, using MATLAB is less than 13 seconds for a single face.

For the evaluation of the iris center position and its radius, it is not very easy to determine the ground truth because first of all, even for a simple perfect circle it is not easy to find the accurate center manually. Measuring the "width" of the iris to find the radius is also defective since the diameter should be measured through the exact center. Hence, manual marking or measurement of these features can yield to incorrect evaluations. For this reasons, the results are examined visually for iris extraction which are given in Table 1.

Table 1. Success rates for iris localization

Method	Threshold	Success rates
[6]	-	94.82%
[7]	-	94%
Our method –without the eye region extraction module	5 pixels	95.23%
Our method –with the eye region extraction module	5 pixels	100%

On the other hand, since the eye corners are not very well defined, visual inspection is not an option. Hence, the corners are marked manually to constitute the ground truth. The error for the eye corners are calculated in two different manners.

Firstly, the error is taken to be the Euclidean distance between the detected and the manually labeled corner. Secondly, as suggested in [11], the error is defined as this distance divided by the standard Euclidean distance between two inner eye corners. But since this "standard" distance is not revealed in [11], it is taken to be:

- the distance between two inner eye corners in that image
- the mean distance between two inner eye corners of all images

For the first error definition, the results are given in Fig. 6, where the threshold considered for accurate detection is defined from 1 to 10 pixels. This error is calculated after the images are scaled according to the distance between two iris centers, to be comparable with [12], in which this distance is fixed to 60 pixels. In Table 2, the results are listed.
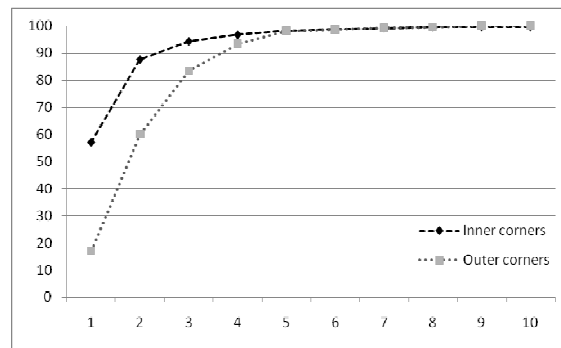

Fig. 6. Eye corner detection rates for different thresholds – Approach #1

For the second one, both approaches are evaluated. According to these results, it is revealed that using a constant to scale the error can be misleading. As can be seen in Fig. 7, the success rates are seemed to be higher when the mean distance between two inner eye corners is used instead of the real distance for each image itself. The second approach is much more informative and hence presented here to be used in further comparisons. In the graph, the error threshold is scaled from 1% to 15%.
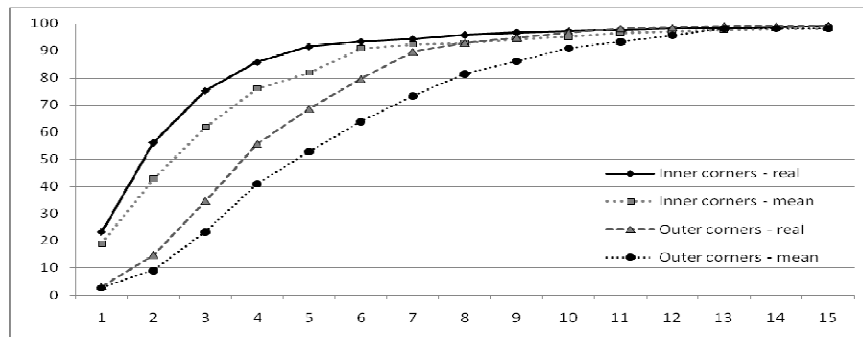

Fig. 7. Eye corner detection rates for different thresholds – Approach #2

Table 2. Success rates – Approach #1

Method	Threshold	Success rates for inner corners	Success rates for outer corners
[12]	4 pixels	96.89%	94.89%
Our method	4 pixels	96.67%	93.33%

Table 3. Success rates – Approach #2

Method	Threshold	Success rates for inner corners	Success rates for outer corners
[11]	5%	95.7%	93%
Our method (using mean)	5%	91.43%	68.57%
Our method (using real)	5%	81.90%	52.86%
Our method (using mean)	10%	97.14%	96.67%
Our method (using real)	10%	95.24$	90.95%

In both ways, it is shown that the algorithm performs better for the inner eye corners. This is because closer to the inner eye corners, the eyelid contours are more prominent than the ones around the outer eye corners.

6   Conclusion

In this paper, an accurate method for automatic detection of eye features is presented. Firstly, the iris position and radius is extracted by using the edges in the non-skin region in the upper half of the face and then, refined by using vertical edges only in a smaller window. Afterwards, also with the help of the previously extracted iris, edges which are formed by the eyelids are detected and two curves for upper and lower eyelids are fitted. Finally, the intersections of these two curves are labeled as the eye corners. Experimental results demonstrated that high accuracy can be achieved with the proposed algorithm. In Fig. 8, some extraction results are illustrated.
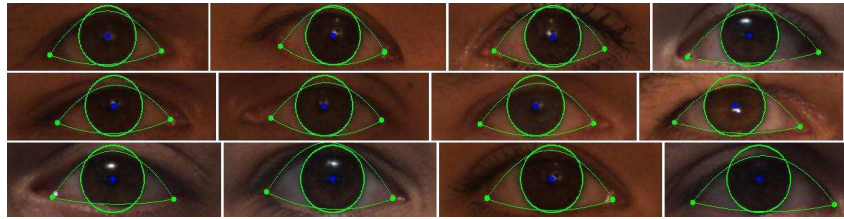

Fig. 8. Some extraction results for iris center and border, eyelid contours and eye corners

References

1 Brunelli, R.; Poggio, T., "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol.15, no.10, pp.1042-1052, Oct 1993

2 Zhonglong Zheng, Jie Yang, Limin Yang, A robust method for eye features extraction on color image, Pattern Recognition Letters, Volume 26, Issue 14, 15 October 2005, Pages 2252-2261, ISSN 0167-8655.

3 Khairosfaizal, W.M.K.W.M.; Nor'aini, A.J., "Eyes detection in facial images using Circular Hough Transform," Signal Processing & Its Applications, 2009. CSPA 2009. 5th International Colloquium on , vol., no., pp.238-242, 6-8 March 2009

4 Pardas, M.; , "Extraction and tracking of the eyelids," Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on , vol.6, no., pp.2357-2360 vol.4, 2000

5 Chan Woo Park; Jeong Min Kwak; Hyun Park; Young Shik Moon; , "An Effective Method for Eye Detection Based on Texture Information," International Conference on Convergence Information Technology, 2007, pp.586-589, 21-23 Nov. 2007

6 Yepeng Guan, "Robust Eye Detection from Facial Image based on Multi-cue Facial Information," Control and Automation, 2007. ICCA 2007. IEEE International Conference on , vol., no., pp.1775-1778, May 30 2007-June 1 2007

7 Kuo, P.; Hannah, J., "An improved eye feature extraction algorithm based on deformable templates," Image Processing, 2005. ICIP 2005. IEEE International Conference on , vol.2, no., pp. II- 1206-9, 11-14 Sept. 2005

8 Peng Wang; Green, M.B.; Qiang Ji; Wayman, J., "Automatic Eye Detection and Its Validation," IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops, pp.164-164, 25-25 June 2005

9 Zhi-Hua Zhou, Xin Geng, Projection functions for eye detection, Pattern Recognition, Volume 37, Issue 5, May 2004, Pages 1049-1056, ISSN 0031-3203.

10 G. Feng. Variance projection function and its application to eye detection for human face recognition. Pattern Recognition Letters, 19(9):899–906, July 1998.

11 Xia Haiying; Yan Guoping, "A Novel Method for Eye Corner Detection Based on Weighted Variance Projection Function," Image and Signal Processing, 2009. CISP '09. 2nd International Congress on , vol., no., pp.1-4, 17-19 Oct. 2009

12 Cui Xu; Ying Zheng; Zengfu Wang, "Semantic feature extraction for accurate eye corner detection," Pattern Recognition, 2008. ICPR 2008. 19th International Conference on , vol., no., pp.1-4, 8-11 Dec. 2008

13 Lijun Yin, Anup Basu, Integrating active face tracking with model based coding, Pattern Recognition Letters, Volume 20, Issue 6, June 1999, Pages 651-657, ISSN 0167-8655.

14 Vezhnevets V., Degtiareva A., "Robust and Accurate Eye Contour Extraction". Proc. Graphicon-2003, pp. 81-84, Moscow, Russia, September 2003.

15 Tse, K.W.; Lau, W.H.; Leung, S.H.; Liew, A.W.C., "Eye extraction using spatial fuzzy clustering method," TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering , vol.1, no., pp. 515- 518 vol.1, 28-31 Oct. 2002

16 MATLAB Central - http://www.mathworks.com/matlabcentral/
Program for Color Image Segmentation – Athi Narayanan S, K.S.R. College of Engineering, Erode, Tamil Nadu, India.

17 Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, L. Akarun, Bosphorus Database for 3D Face Analysis, The First COST 2101 Workshop on Biometrics and Identity Management (BIOID 2008), Roskilde University, Denmark, May 2008.