Eye Detection Algorithm on Facial Color Images

Jalal A. Nasiri[α], Sara Khanchi[β], Hamid R. Pourreza[α]

Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran[α]
Department of Computer Engineering, University of Yazd, Yazd, Iran[β]

Ja_na989@stu-mail.um.ac.ir, s.khanchi@yazduni.ac.ir, hpourreza@um.ac.ir

Abstract

In many application suck as face detection or recognition a major phase would be eye detection. In addition, its wide use as a part of serious applications, made it an important task should be worked on. Using color characteristics is a useful way to detect eyes. We use special color space, YCbCr which its components give us worthwhile information about eyes. We make two maps according to its components and merge them to obtain a final map. Candidates are generated on this final map. We apply an extra phase on candidates to determine suitable eye pair. The extra phase consists of flexible thresholding and geometrical tests. Flexible thresholding makes generating candidates more carefully and geometrical tests allow proper candidates to be selected as eyes. Simulation results on CVL and Iranian Databases showed this phase improved the correct detection rate by about 12.4% and reach 98.5% success rate on the average.

Keywords: Eye detection, Color images, Lighting condition, Facial feature map.

1. Introduction

Eye detection is a crucial step in many applications such as face detection/recognition, face expression analysis, gaze estimation, criminal investigation, human interactions and surveillance systems [2, 3, 4]. Existing works in eye detection can be classified into two major categories: traditional image-based passive approaches and the active IR based approaches. The former uses intensity and shape of eyes for detection and the latter works on the assumption that eyes have a reflection under near IR illumination and produce bright/dark pupil effect.

The traditional methods can be broadly classified into three categories: template based methods [8,9], appearance based methods [10, 11] and feature based methods [12, 13]. Our approach is considered to be in first category.

Color is one of the useful features used for eye detection. Thilak et al. [14] proposed an algorithm which by three levels detects eyes. First they localized eye candidates by simple thresholding on HSV color space and normalized RGB color space sequentially. Then it is followed by connected component analysis to determine spatially connected regions and reduce the search space to determine the eye pair windows. Ultimately, the mean and variance projection function is applied to validate the presence of eye in each window. Lin et al. [15] proposed an algorithm which uses HSI color space to extract skin color pixels and use region growing algorithm to group these pixels. Then by means of Face Circle Fitting (FCF) method, they detect face region and thereafter apply Dark-pixel Filter (DFP) to identify eye candidates. At last, they use geometric relation to find eye positions. Gargesha et al. [16] combine the techniques of chrominance and luminance and curvature analysis to compute eye maps. The exact position of eyes could be determined using either PCA or Random transform.

We construct Eye Maps and by combining them, determine eye candidates from the final EyeMap. Eye Map is obtained from a facial image that is transformed into YCbCr color space [1]. The two highest peaks (brightest regions) in Eye Map are supposed to be eyes [6].

Our simulation results showed that two highest peaks don't always correspond to eyes, i.e. input image

978-0-7695-3136-6/08 $25.00 © 2008 IEEE
DOI 10.1109/AMS.2008.55

is noisy or under poor lighting conditions. An extra phase is designed to overcome these situations. In this phase, the bright regions that satisfy some special features are considered as eye pair. Experimental results showed this phase improved detection rate saliently.

2. Algorithm of Eye Detection

We first build two separate eye maps from facial image, EyeMapC from the chrominance components and EyeMapL from the luminance component. These two maps are then combined into a single eye map, EyeMap. The facial image should be frontal and not occluded by objects like glasses, mask and so on. Also both eyes should be visible in input image so head rotation at most 30° around vertical axis and 10° around horizontal axis is acceptable.

The flowchart of whole algorithm is shown in Fig.1.

2.1. EyeMapC

The main idea of EyeMapC is based on characteristics of eyes in YCbCr color space which demonstrates that eye regions have high Cb and low Cr values [6]. It is constructed by:

$$\text{EyeMapC} = \frac{1}{3}\left\{\left((\mathrm{C_b})^2 + (\overline{Cr})^2 + (\frac{C_b}{Cr})\right)\right\}$$

Where $(C_b)^2$, $(\overline{Cr})^2$, (C_b/C_r) all are normalized to the range [0 1] and $(\overline{Cr})^2$ is the negative of Cr (i.e., 1-Cr). This formula is designed to brighten pixels with high Cb and low Cr values. $(C_b)^2$ emphasizes pixels with higher Cb value and causes pixels with lower Cb value become weaker, also (C_b/C_r) results in pixels with low Cr become brighter. Finally (C_b/C_r) component completes our idea that eye regions have high Cb and low Cr values.

The 1/3 scaling factor is applied to ensure that the resultant EyeMapC stays within the range of [0 1]. Eventually, we perform histogram equalization on it to obtain final EyeMapC. The process of EyeMapC construction is shown in Fig.2.

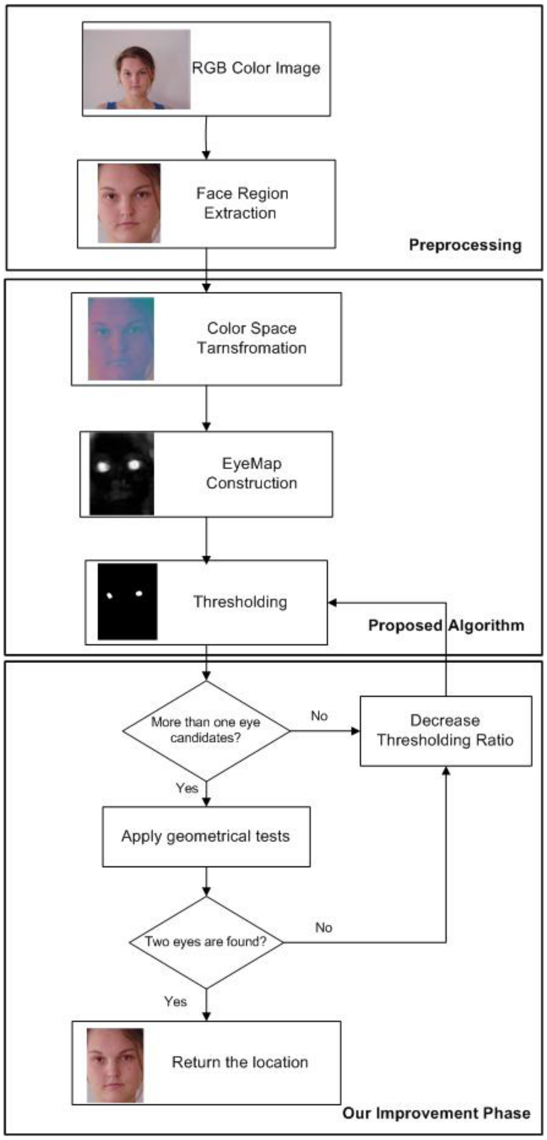


Figure **1:** The Whole Algorithm

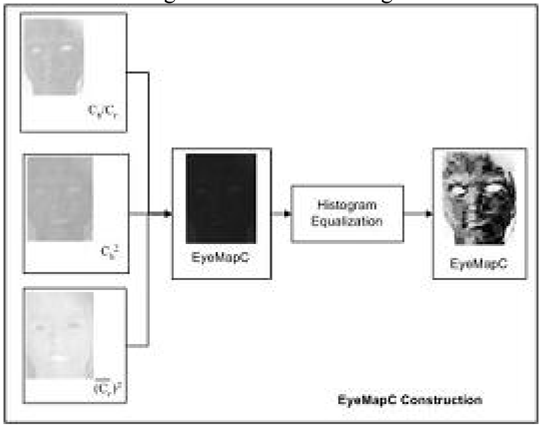


Figure 2: EyeMapC

2.2. EyeMapL

Since the eyes usually contain both dark and bright pixels in the luma component, grayscale morphological operators (e.g., dilation and erosion) [5] can be designed to emphasize brighter and darker pixels in the luma component around eye regions. We use grayscale dilation and erosion with a hemispheric structuring element to construct eye map from the luma as follows:

$$\text{EyeMapL} = \frac{Y(x, y) \oplus g(x, y)}{Y(x, y) \otimes g(x, y)}$$

Where the grayscale dilation $\oplus$ and erosion $\otimes$ operations on a function: $f : F \subset R^2 \rightarrow R$ using a structuring function $g : G \subset R^2 \rightarrow R$ are defined in [5].

2.3. Eye Map

After constructing EyeMapC and EyeMapL, we multiply them to obtain the final 'EyeMap', i.e., EyeMap= (EyeMapC) ***AND*** (EyeMapL), Fig.3. The resulting eye map is then dilated, masked, and normalized to brighten both the eyes and suppress other facial areas. The locations of the eye candidates are estimated and then refined using thresholding and binary morphological erosion on this eye map.

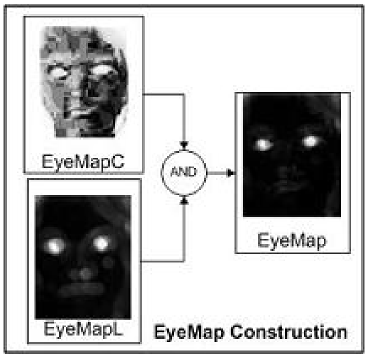


Figure 3: EyeMap

3. Improvement Phase

As discussed before, two brightest regions in EyeMap are not always eyes [17]. We proposed a novel approach to ensure that two selected regions are eyes, not necessarily two brightest ones. Hence we choose among eye candidates those two regions that satisfy our conditions. We utilized ***flexible thresholding*** and ***geometrical tests*** to design our approach which in following we describe them.

Flexible Thresholding

The minimum value each pixel must have to be considered as white (255) after thresholding is determined by a parameter called thresholding ratio which is different in images under various lighting conditions. Setting the thresholding ratio is the bottleneck of our solution. Adjusting the threshold ratio so much high in some images, results in eye regions not to be considered as eye candidates. In contrast, selecting the threshold ratio low causes firstly number of eye candidates increase and finding eyes among them becomes difficult, secondly regions combine with each other and cause not detecting the exact position of eyes. We reached this flexibility by combining iterative thresholding and geometrical tests which are explained in improvement algorithm.

Geometrical Tests

Eyes have some features in faces. We extract features and design special tests to verify which candidates are eyes. The geometrical tests include followings:

- ***Eyes-Centre Distance Test****:* We calculate distance of eyes from the centre of face. Both the distance between eye and the centre of face are almost the same and must not exceed each other by 30%. Our observation shows the two distances are found very closely matched.
- ***Eye Pair Distance Test****:* The distance between the eye pair must be more than *Eyes-Centre Distance.*
- ***Eye Angle Test****:* According to the structure of face, two eyes cannot be located in one side of face. In this test we examine eye pair candidates to be in two side of face.
- ***Eye Shape Test:*** As eye shape is circular, we test candidates not to be so thin and long. For this purpose we compute ***roundness ratio***. The two selected eyes must be more than **0.7** in ***roundness ratio***. Also those thin and long candidates would be discarded in this test.

Improvement Algorithm

Our suggested algorithm is composed of iterative thresholding and geometrical tests. We set thresholding ratio based on maximum pixel value in the facial image. Experimental results demonstrate that selecting too high thresholding ratio is not proper and sometimes it is dangerous to start with it. We found out 0.7*MaxValue is the best to start with. Then geometrical tests are applied on eye candidates obtained after thresholding. If two regions found that satisfy all the tests, they considered as eyes and algorithm finishes, else algorithm restarts by lowering the threshold ratio.

The next ratio is obtained by previous minus 0.1 (ratio= previous ratio-0.1). This iterative task would be continued till both eyes are detected by algorithm. The Flowchart of improvement algorithm is shown in Fig.4. As our observations showed, when the light distribution is not unified over face, the number of iterations increases.

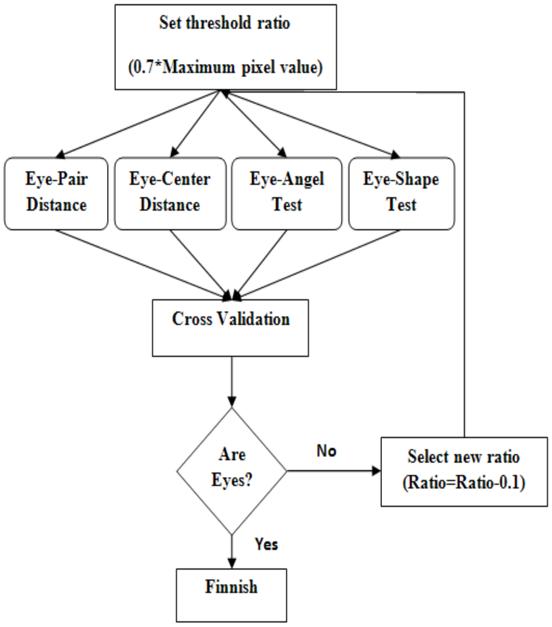


Figure 4: Improvement Algorithm

4. Experimental Results

This section provides simulation results to evaluate our algorithm with and without improvement phase. We apply our algorithm on CVL [7] and Iranian databases. When the detection is found, a cross is used to mark the eyes. Summary of the detection results (including the number of images, detection rates and average CPU time for processing an image) on the CVL and Iranian databases are presented in Table.3 and Table.4, respectively. The detection rate is computed by the ratio of the number of correct detection to that of all the images tested. Sample of detections on CVL and Iranian databases are demonstrated in Fig.5 and Fig.6.

CVL Database

CVL database consists of head and shoulder images taken from 114 people in 7 kinds of expressions. Among 7 images taken from a person, 3 of them are suitable for our purpose. These three photos are frontal view and with different expressions: serious, smile and grin.

Iranian Database

Iranian database consists of head and shoulder images taken from 50 people. Images in Iranian database are taken under various lighting conditions.

Table 1: Results on CVL database without improvement phase

Expression	Serious	Smile	Grin	Total
No. of image	110	110	110	330
Data Rate (%)	90	86.36	81.81	86.06
Time (sec):average	0.69	0.70	0.71	0.70

Table 2: Results on the Iranian database without improvement phase

Gender	Female	Male	Total
No. of image	28	22	50
Data Rate (%)	85.71	86.36	86
Time (sec):average	1.21	1.54	1.375

Table 3: Results on CVL database with improvement phase

Expression	Serious	Smile	Grin	Total
No. of image	110	110	110	330
Data Rate (%)	100	98.18	97.27	98.48
Time (sec):average	0.72	0.71	0.74	0.724

Table 4: Results on the Iranian database with improvement phase

Gender	Female	Male	Total
No. of image	28	22	50
Data Rate (%)	96.42	95.45	96
Time (sec):average	1.23	1.56	1.395

5. Conclusions

In this paper we have presented an eye detection algorithm for color image. Our method detects eyes in face image which is extracted over the entire image. We proposed flexible thresholding and geometrical tests and applied on eye candidates. Our simulation results demonstrate the effectiveness of improvement phases have increased the correct detections by about 12.4% and we reach the detection rate about 98.5% in total.

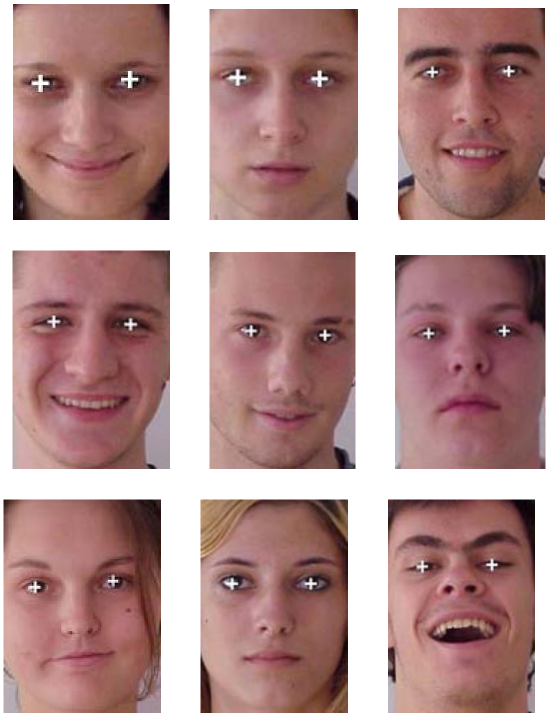
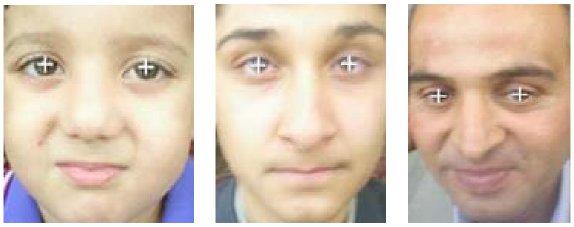


Figure 5: Sample of CVL Detection



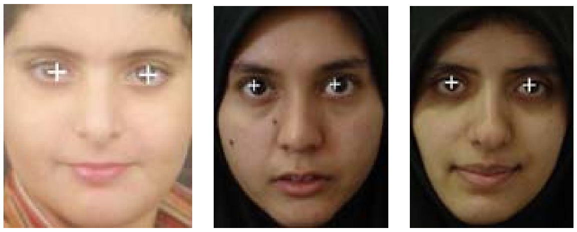


Figure 6: Sample of Iranian Detection

6. References

[1] Douglas Chai and King N.N. Face segmentation using skin color map in videophone application. IEEE Trans. Circuits Systems for video Technology, 9(4), June 1999.

[2] I. Craw, D. Tock, and A. Bennett. Finding face features. In Proc. Second European Conf. Computer Vision, pages 92–96, 1992.

[3] Santa Ana Police Department. Algorithmic image matching (aim): Project analysis for santa ana police department. Research, April 2000.

[4] H.P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In Proc. First Intl Workshop Automatic Face and Gesture Recognition, pages 41–46, 1995.

[5] P.T. Jackway and M. Deriche. Scale-space properties of the multiscale morphological dilation-erosion. IEEE Trans. Pattern Analysis and Machine Intelligence, 18:38–51, Jan. 1996.

[6] Hsu Rein-Lien, A. M. Mohamed, and K. J. Anil. Face detection in color images. IEEE Trans. Pattern Analysis and Mechine Intelligence, 24(5), 2002.

[7] Franc Solina, Peter Peer, Samo Juvan, and Jure Kovac. Color-based face detection in the ”15 seconds of fame” art installation. In Mirage 2003, Conference on Computer Vision / Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects, pages 38–47, INRIA Rocquencourt, France, March 2003.

[8] G.C. Feng and P.C. Yuen. Multi-cues eye detection on gray intensity image. Pattern Recognition, (34):1033–1046, 2001.

[9] W. min Huang and R. Mariani. Face detection and precise eyes location. In Conf on Pattern Recognition (ICPR00), 2000.

[10] Jeffrey Huang and Harry Wechsler. Eye detection using optimal wavelet packets and radial basis functions (rbfs). Pattern Recognition and Artificial Intelligence, 13(7):1009–1026, 1999.

[11] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR94), Seattle,WA, 1994.

[12] S. A. Sirohey and Azriel Rosenfeld. Eye detection in a face image using linear and nonlinear filters. Pattern Recognition, 34:1367–1391, 2001.

[13] S. Kawato and N. Tetsutani. Detection and tracking of eyes for gaze-camera control. In Proc. 15th Int. Conf. on Vision Interface, 2002.

[14] R. Thilak, S. Kumar Raja, and A.G. Ramakrishnan. Eye detection using color cues and projection functions. In Proceedings 2002 International Conference on Image Processing ICIP, volume 3, Rochester, New York, USA, 2002.

[15] Daw-Tung Lin and Chen-Ming Yang. Real-time eye detection using face circle fitting and dark-pixel filtering. IEEE International Conference on Multimedia and Expo (ICME), 2004.

[16] M. Gargesha and S. Panchanathan. A hybrid technique for facial feature point detection. Fifth IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI'02), 2002.

[17]Nasiri J.A., Khanchi S. and Pourreza, H.R. . Eye Detection in Color Images. 4th *Iranian Conf. on Machine Vision and Image Processing (MVIP07)*; 2007.