Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 4, April 2013, pg.95 – 100

RESEARCH ARTICLE

Elliptical Iris Detection using Morphological Operations

Prakash. J. Kulkarni[1], Vidya H. Kopanar[2]
[1]CSE Department, Walchand College of Engg., Sangli, India
[2]CSE Department, Walchand College of Engg., Sangli, India

[1] pjk_walchand@rediffmail.com; [2] vidyakopanar@gmail.com

Abstract— This paper presents a method to detect an iris to operate in 3D coordinate system. As per projective geometry, circle in 3D is projected as an ellipse in 2D. Hence, iris is detected as an ellipse instead of a circle. Morphological operations of light computations are used to detect an iris, which reduces time to detect an iris. Eye images with high resolution provide more number of pixels giving accuracy. Such detected ellipse can be further used to map to a circle in 3D using calibrated camera.

Key Terms: - Morphological operations; Ellipse fitting; Edge following; Region filling; Edge detection

I. INTRODUCTION

Iris detection is a well-developed research area in computer science. It has large numbers of applications in the field of image processing such as passport verification, iris recognition, human machine interface, person identification, passwords security, etc. Research in image processing is leading towards working with 3D coordinates. Many applications are being developed which works with 3D coordinates. It is very easy to obtain 3D coordinates of an object once parameters of camera are measured using camera calibration. This could also help to improve an accuracy of a system.

As technology is developing day-by-day and being used in many applications, there is a need to have more natural ways to interact with the system.

This leads to developing applications such as hand gesture recognition, eye tracking, etc. Eye tracking along with gaze estimation estimates a point where the person is looking on the screen [1]. Such information can be used to initialize certain events based on some constraints. Performance of such system highly depends on accuracy in detection of an iris.

In applications, based on iris detection, accurate iris detection is a crucial task as it affects the performance of the system. Iris detection using deformable template is a well-known method [2]. However, performance of this method depends upon initial position of detection. It is efficient only if starting location of iris detection is near to the eye.

Also, it needs weight values to be calculated manually for some energy terms. In presented method, morphology-based image processing operations are used to detect an iris [3]. Dilation and erosion are two basic morphological operations which are used in different combinations iteratively to serve the purpose. Such morphological operations are computationally inexpensive.

The presented method uses eye images captured with zoom-in camera as an input to the system. This provides more number of pixels to process which ultimately results into improved accuracy. Eye image is segmented first by using skin color to separate eye from the skin color. Iris is, then, detected by applying morphological operations on the image of the eye. Processed image provides image points which, then, can be fitted to an ellipse using ellipse fitting method. Many applications are moving towards processing in 3D coordinates, instead of 2D co-ordinates. As per projective geometry, a circle or sphere in 3D is projected as an ellipse in 2D image. Iris is circular in 3D; hence, in this method, instead of a circle, an iris is detected as an ellipse which is a projection of circular iris in 3D in 2D.

Such detected elliptical iris can be, then, further used in application.

II. METHODOLOGY

An eye image is captured with a zoom-in camera of high resolution. This provides iris images with more number of pixels. Overview of iris detection is given in fig 1.

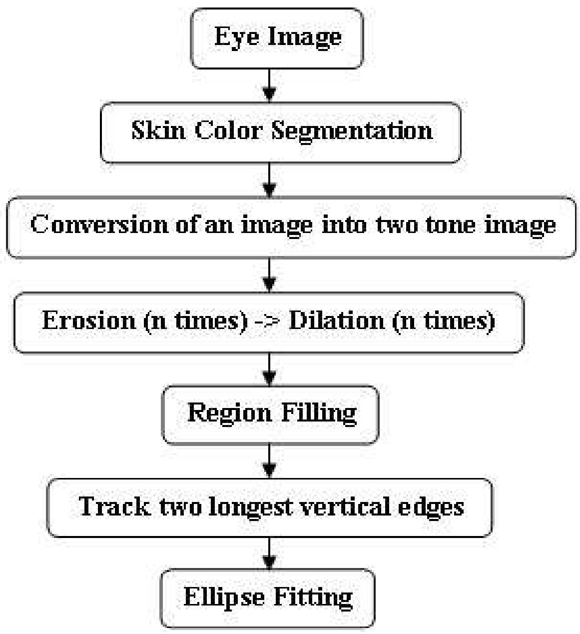

Fig 1: Overview of Iris detection

A. Skin Color Segmentation

Wrinkles near eye are usual. Such wrinkles are outputted as an edge if edges are computed without undergoing skin color segmentation. This is considered as a noise. Hence, skin color segmentation becomes essential.

RGB color space is not efficient to detect a skin color. Results of segmentation of an image based on hue color are found to be the best for detection of skin color [4]. Hue color is separated from brightness and it is a pure color component. Hence, hue satisfactorily separates out skin color.

Hue value can be computed using R, G and B values. Hue component of a color is computed using following equations 1 and 2.

$$hue = a\cos\left(\frac{((R-G)+(R-B))}{(2*sqrt((R-G)*(R-G)+(R-B)*(G-B)))}\right) \quad \textbf{(1)}$$

$$hue = hue*180/3.14 \quad \textbf{(2)}$$

The condition for a color to be a skin color is given in equation 3.

$$\begin{aligned}&(hue < 45 \,\&\&\, hue > 10)\,OR\\&\left(\begin{aligned}&((ar - a*dr) < R)\,\&\&\,(R < (ar + a*dr))\\&\&\&\,((ag - a*dg) < G)\,\&\&\,(G < (ag + a*dg))\end{aligned}\right)\end{aligned} \quad \textbf{(3)}$$

Where, a=1, ar=0.4408, dr=0.0713, ag=0.3070, dg=0.0271.

All pixels which fall under above range are separated as a skin.

B. Segmentation

After completing skin color segmentation, resulted image consists of:

1. Black-coloured portion of an eye i.e. iris, eyebrows and eyelids.
2. white-coloured portion of an eye i.e. sclera
3. white part segmented in skin color segmentation

Purpose of this segmentation stage is to separate iris region from remaining part of an image. The threshold for segmentation is computed by analysing intensity histogram [5]. An image, to be threshold, is assumed to have two classes, foreground and background. For each intensity value, spread of number of pixels, for each class, is analysed and an intensity value, for which sum of foreground and background spreads is at its minimum, is selected as threshold. Based on this threshold value, image is segmented (Iris region is made white and non-iris region is made black).

C. Removal of eyelids

As two longest vertical edges are going to be used to detect an iris, edges occurring due to eyelids may cause problem. Hence, eyelids are removed by applying erosion operator for multiple times. This is followed by dilation for the same number of times to retain the shape of an iris.

D. Region Filling

Lightning conditions do matter while taking a picture. It may happen that iris part contains reflection of a light. Such reflection leads to forming an edge inside iris. Hence, it becomes necessary to fill that portion. Region filling is performed by applying dilation operator for multiple times to fill holes within iris. This is followed by erosion for the same number of times to retain the shape of an iris.

E. Edge Detection

Leading towards the edge detection, canny edge operator is used [6]. Canny edge detection algorithm consists of 5 steps:

1. Noise reduction
2. Gradient computation
3. Non-maximum suppression
4. Double thresholding
5. Hysteresis

An input image for edge detection is convolved with Gaussian filter. It produces a blurred version of an input image which may contain noisy pixels with no significance. Gradients in both horizontal G_x and vertical G_y directions are measured with edge detection operator. The final gradient and angle of an edge are computed as per equation 4 and 5 respectively:

$$G = \sqrt{G_x^2 + G_y^2} \tag{1}$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{2}$$

This results into blurred edges. To convert these blurred edges in image of gradient magnitudes to sharp edges. This is done by preserving all local maxima and deleting everything else.

Double thresholding step makes use of two thresholds, viz. T_1 and T_2 where $T_1 < T_2$. All pixels with intensity value below T_1 are suppressed. All pixels with intensity between T_1 and T_2 are considered as weak edges and all pixels with intensity above T_2 are considered as strong edges. Hysteresis step preserves all strong edges and only those weak edges which are connected to strong edges.

F. Filtration of Vertical Edge and Edge Following

With the orientation of an edge, all the vertical edges are tracked and only two vertical longest edges are retained with edge following technique [7].

For edge following algorithm, 8-connectivity is considered. To find all edges, image is scanned from left-to-right and top-to-bottom. Pixel P in an image is considered as a pixel under consideration. Pixel Q is just a previous pixel to pixel P in scanning an image.

• Start with initial values of P and Q.

• Given P and Q values, find next pair of P and Q.

Possible values of Q are 8-neighborhood of $P : \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8\}$

Let R_i be first R in this sequence which is unmarked edge pixel.

Add pixel P to list $L = \{\ \}$.

Set $P = R_i$ and $Q = R_{i-1}$.

• Continue with step 2 until no unmarked pixel is found.

• Continue with further scanning.

Edges are tracked and edge lengths are calculated in terms of pixels. Only first and second largest edges are preserved and remaining are deleted. These two vertical edges usually come up as two sides of an iris.

G. Iris Detection as an Ellipse

With the reliable iris edges, iris contour is fitted to an ellipse. An ellipse fitting algorithm [8], is used to get an accurate estimation of an ellipse.

III. RESULTS

Program is run on Ubuntu 12.04LTS with Intel Optiplex 790. Methodology is implemented in CPP with the help of OpenCV library.

Eye images of size 640 by 480 are used as input images. These images are captured using high resolution camera. Use of such camera provides us images with more number of pixels and hence, more pixels will be available to process which could result in improving results.

Iris is going to be found within particular region of an eye image. It would be efficient to compute such area where possibility of an iris is assured. Hence, search space must be computed first for further detection of an eye.

Once search space for an iris is located, iris detection can be applied over that space and processing time can be reduced.

A. Computing Search Space

To locate iris location, search space is computed using very first frame in which iris is at centre. It reduces processing time for further detection of an iris. The constraint, while computing search space, is iris must be at centre.

Eye image is taken as input and undergoes skin color detection. Resulted image is segmented to separate out sclera portion of an eye i.e. a white portion of an eye. Such segmented image contains noise. To remove noise, morphological erosion operator of size 3 by 3 is applied multiple times. To regain iris area, dilation operator of size 3 by 3 is applied for same number of times.

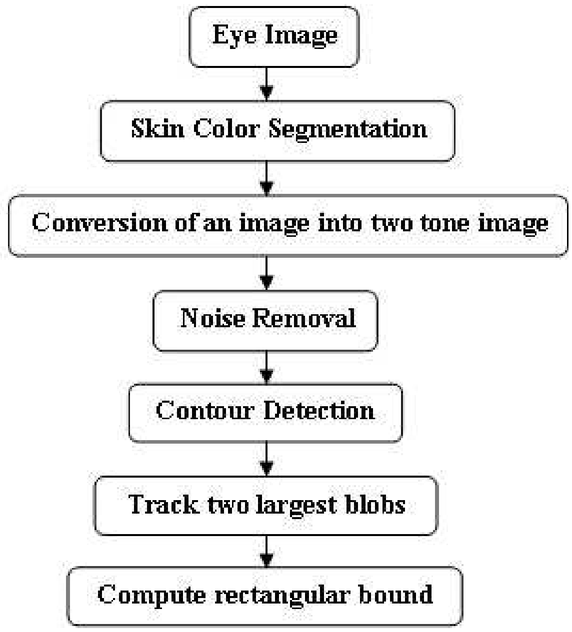

Fig 2: Computation of search space to locate iris

Contours are obtained from resulted image and first two largest contours are selected. Using contour points**,** bounding rectangular search space can be computed. Then, for every person, iris can be detected within this search space provided, head movement is not allowed.

In Fig 3, (a) shows an eye image, (b) is an image after segmentation. Threshold is applied on skin colour segmented image, as shown in (c). (d) Shows an image after removing noise. (e) Contains two largest blobs which appear as white portion of an eye, i.e. sclera. (f) Displays contours of two largest blobs. Computed search space is as shown in image (f).

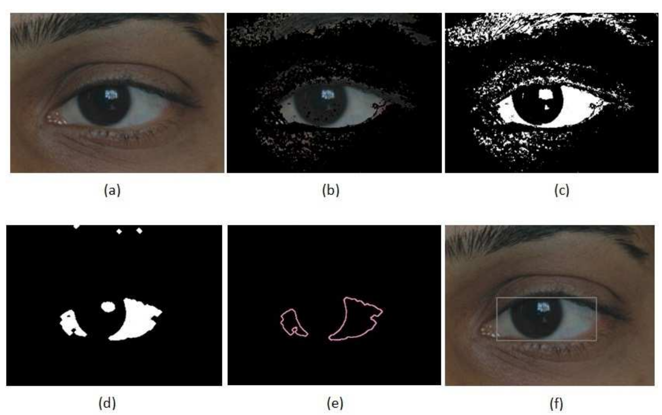

Fig. 3 Results for computation of searched space

B. Iris Detection

Iris is detected using morphological operations. Dilation and erosion are two major operations which are used with 3 by 3 operators. Skin color detection helps to reduce noise which can appear in further processing. Image is segmented with the help of threshold to convert it into two-tone image, which generates iris, eyelids and eye-brows as a white portion.

Iris is detected using vertical edges. Clearly, eyelids may cause false detection. Hence, it becomes necessary to remove eyelids. It also reduces noise in an image. Region filling is applied to fill the holes within an iris. Resulted image, then undergoes edge detection and vertical edges are retained and remaining is discarded. Such vertical edges are followed within the search space and only two longest edges are considered to detect an iris.

These two edges are used for ellipse fitting. Such fitted ellipse, then, can be used to obtain the 3D location of an iris with the help of camera calibration. In Fig 4, (a) is an original image. This image is segmented using skin color which is resulted as (b). Threshold is applied, further, resulting binary image containing eye-brows, eyelids and iris as white. Iris is going to be detected using vertical edges. So, it becomes necessary to remove eyelids. (d) Shows eye image with removed eyelids. Due to illumination, iris comes with holes. These holes are filled region filling, as shown in (e). Vertical edges are determined as shown in (f) and those within search space are tracked. Only first two largest vertical edges are considered as shown in (h). Ellipse is fitted to such image points and iris is detected as shown in (g), (h).

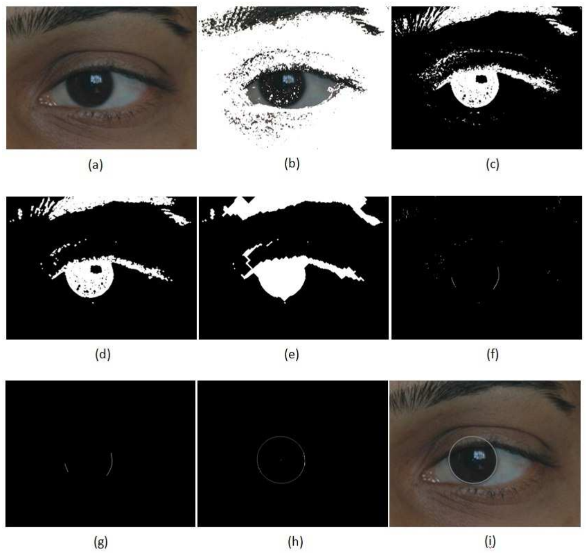

Fig. 4 Results for Iris detection

Compared to other methods of iris detection such as, deformable template, this iris detection method is simple and based on light computations.

Accuracy of presented iris detection method is checked using synthetic dataset. This synthetic dataset is prepared manually for which centre of iris is known. Results generated by executing presented method are compared with already known true values. The results are found to be nearly 70% accurate.

IV. CONCLUSIONS

Morphological operations are not compute-sensitive. Hence processing an iris takes less time. Iris is detected as an ellipse instead of a circle, as a projection of circular ring in 3D. Such detected iris can be further used in applications which work with 3D world co-ordinate system.

REFERENCES

[1] Jian-Gang Wang, Eric Sung, Ronda Venkateswarlu, Eye Gaze Estimation from a Single Image of One Eye, ICCV 2003: 136-143.
[2] Yuille, A., Haallinan, P., Cohen, D.S., Feature extraction from faces using deformable templates, Proc. IEEE Computer Vision and Pattern Recognition, pp. 104–109 (1989).
[3] http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm
[4] Lamiaa Mostafa and Sherif Abdelazeem, Face Detection Based on Skin Color Using Neural Networks, journal of Network and Computer Applications (2002), pp. 1-3, December-2002.
[5] Otsu, A threshold selection method from gray-level histogram, IEEE Transactions on Systems, man and Cybernetics, vol. SMC-9, 1979, pp. 62–66.
[6] J. Canny (1986), A computational approach to edge detection, IEEE Trans. Pattern Analysis and Machine Intelligence, vol 8, pages 679-714.
[7] http://www.slideshare.net/kiara1011000/edge-following-algorithm-chiara-galdi
[8] Andrew W. Fitzgibbon, R. B. Fisher. A Buyer's Guide to Conic Fitting, Proc.5th British Machine Vision Conference, Birmingham, pp. 513-522, 1995.

