J. Basic. Appl. Sci. Res., 3(2s)790-796, 2013
© 2013, TextRoad Publication

ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

A Parallel Method for Eye Detection and Tracking with Complex Background

Mohammad Ali Azimi Kashani 1 Mohammad Reza Ramezanpour Fini 2

[1] Young Researchers Club, Shoushtar Branch, Islamic Azad University, Shoushtar, Iran
[2] Departments of Computer, Science & Research Branch, Islamic Azad University, Tehran, Iran

Received: June 10 2013
Accepted: July 10 2013

ABSTRACT

This paper presents a method for eye detection and tracking using local feature of face. Moreover, we have proposed parallel algorithm to reduce eye detection time. To detect and track eye images with complex background, distinctive features of user eye are used. Face detection, eye region detection, pupil detection and eye tracking is four steps that used in an eye-tracking and detection system. We Used color characteristics to detect face. First, face region separate from the rest of the image using parallel mixture of Gaussian. This will cause the images background to be non effective in our next steps. EREW PRAM model used for this section. For decrease the computational complexity and ignoring some factors such as bread we used the horizontal projection obtained from face region, to separate a region containing eyes and eyebrow. Finally, in proposed method points with the highest values k of are selected as the eye candidate's. The eye region is well detected among these points. Color entropy in the eye region is used to eliminate the irrelevant candidates. In the next step, we perform eye tracking with Kalman filter. In the proposed method, eye detection and tracking are applied on testing sets, gathered from different images of face data with complex backgrounds. It is demonstrated by experimental results that the proposed algorithm can detect 98.7% correct while reducing computational complexity.

KEYWORDS: Parallel algorithm, Eye detection, Eye tracking.

1. INTRODUCTION

Automatic tracking of eyes and gaze direction is an interesting topic in computer vision with its application in biometric, security, intelligent human-computer interfaces, and driver's drowsiness detection system. Localization and extraction of eyes are operations requisite for solving problem. In [1] eye localization methods have been classified into five main categories :(1) Shape-based Approaches which described by its shape, which includes the iris and pupil contours and the exterior shape of the eye (eyelids). (2) Feature-Based Shape Methods which explore the characteristics of the human eye to identify a set of distinctive features around the eyes. The limbus, pupil (dark/bright pupil images) and cornea reflections are common features used for eye localization. (3) Appearance-Based Methods which detect and track eyes directly, based on the photometric appearance as characterized by the color distribution or filter responses of the eye and its surroundings. These methods are independent of the actual object of interest and are in principle capable of modeling other objects besides eyes. (4) Hybrid Models which aim at combining the advantages of different eye-models within a single system to overcome their respective shortcomings (5) Other Methods which employing temporal information and active light. In [2], skin-color filter is used to extract face. The eye position is gained by gradient characteristic projection and corresponding conditions setting. In [3] use the Haar-like features to detect the eye. This method trains classifiers with a few thousands of sample view images of object and construct a cascade of classifiers to detect eye rapidly. There are several approaches to reduce eye detection processing time. These approaches can be broadly classified into two groups: software-based and hardware based. The focus in the software-based approach is designing and optimizing an eye detection algorithm for real-time performance. Conversely, in the hardware based approach, the eye detection algorithm is parallelized and implemented on DSP or FPGA to achieve real-time performance. T. D'Orazio et al. propose a real time eye detection algorithm [8]. They use iris geometrical information to determine the eye region candidate and symmetry to select the pair of eyes. Their algorithm consists of two steps. First, the region candidate that contains one eye is detected from the whole image by matching the edge directions with an edge template of the iris. Then, in the opposite region, a search is carried out for the second eye, whose distance and orientation are compatible with the range of possible eye positions. K. Lin et al. proposed a fast eye detection scheme for use in video streams rather than still images [9]. The temporal coherence of sequential frames was used to improve the detection speed. First, the eye detector trained by the AdaBoost algorithm is used to roughly obtain the eye positions. Then, these candidate positions are filtered according to geometrical patterns of human eyes.

2. Skin color segmentation

Since color processing is faster than other face features and color data remain the same as the face turns, skin color will be used to detect face region. One of the choices for the color space without illumination is the chromatic color space YCbCr because the light intensity can be separated from the color space. Hence, the image is first converted from RGB color space into YCbCr. At the time of operation, only Cb and distribution. RGB color space changes into YCbCr by equation (1) [4]:

***Corresponding Author:** Mohammad Ali Azimi Kashani, Young Researchers Club, Shoushtar Branch, Islamic Azad University, Shoushtar, Iran

$$\begin{cases} Y = 0.299R - 0.587G - 0.114B \\ Cb = R - Y \\ Cr = B - Y \end{cases} \quad (1)$$

2.1. Skin color Model Construction

2.1.1. Flush Simple Gaussian Model

A likelihood function could be defined based on the skin color model. This function can determine which skin region a pixel belongs to. Skin color distribution may be reconstructed in YCBCR color space through a normal (Gaussian) distribution with $\sum$ and μ parameters. The multi variable normal distribution of incidental D-dimensional vector x is defined as equation (2):

$$N(x;\mu,\Sigma) = \frac{1}{(2\Pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\exp[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)] \quad (2)$$

Where μ denotes the mean (average) vector and $\sum$ stands for the covariance matrix of vector x . The model parameters are obtained from the instructional data according to equation (3):

$$\mu = \frac{1}{n}\sum_{i=1}^{n} c_i$$
$$\Sigma = \frac{1}{n-1}\sum_{i=1}^{n}(c_i - \mu)(c_i - \mu)^T \quad (3)$$

The Mahalanobis distance from the color vector C to vector μ and the measured covariance $\sum$ denote the belonging of pixel C to the skin color.

2.2. Gaussian Mixture Model

The simple Gaussian model works well in detecting features and distinguishing between different clusters. However, severe calculation errors may occur in this model when the respective components do not vary evenly compared to the mean. Generally stated, Gaussian mixture model is a good approximation of phenomena with natural distribution. In this model, some Gaussian functions are used to model skin color [5]. Gaussian mixture which equals total weight μ is part of Gaussian distribution. The possibility of) the feature vector x to λ_i model is calculated through equation (4).

$$f(x|\lambda_i) = \sum_{i=1}^{M} p_i g_i(x) \quad (4)$$

$g_i(x)$ Denotes the possibility of observing vector x. This observation results from the i th mixture, and p_i is th weight of the i th mixture. Generally, each of the mean vector (μ_i) , covariance matrix (Σ_i) and the mixture weight (p_i) equation (5). Each of the above parameters represents a skin color.

$$\lambda_i = (\mu_i, \Sigma_i, p_i) \quad (5)$$

In fig.1 we are proposed details of parallel face detection.

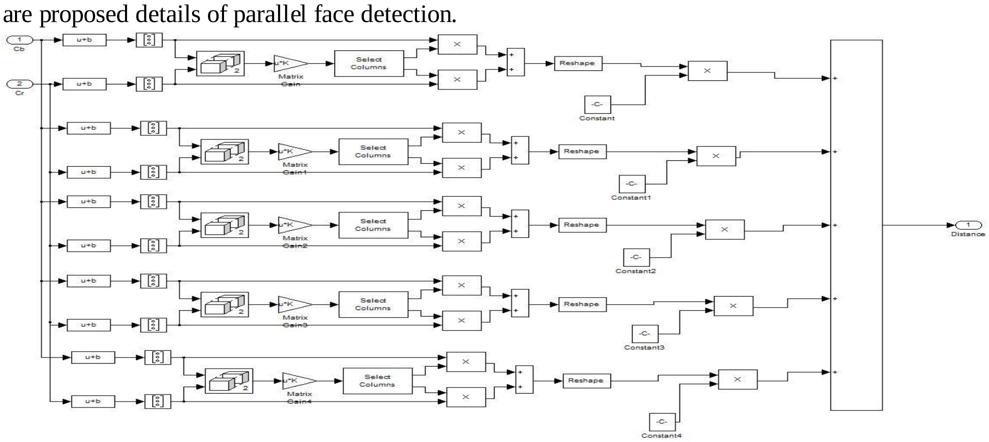


Fig. 1 . Proposed algorithm for face detection.

3. Proposed method

In the proposed method, color information in YCBCR environment and Gaussian mixture model has been used. Since the region above the eyebrows and below the eyes is brighter, the eye region is cropped. This region, which starts above the eyebrows and extends to the region below the eyelids, includes two eyebrows, two eyes, and their characteristic parts like eyelids and eye corners. Applying Harris method, we roughly locate the pupil. Having identified the pupil in the initial frame, we detect the eye region by Kalman filtering. Fig.2 illustrates the proposed method. Since computation of this formula on every single pixel is independent of each other, it can be parallelized as follows. The master processors are responsible for dividing the image into p portions and send each portion to the corresponding Slave processor. Considering p processors, n/p rows of an image can be assigned to one processor.

3.1. Extracting Eye Region

One of the main face regions with a lot of feature is the eye region. Starting above the eyebrows and extending below the eyelids, this region incorporates two eyebrows, two eyes, and their characteristic parts like eyelids and eye corners. To obtain the eye region, the face region is cropped out of the image and then the horizontal projection of the face is worked out.

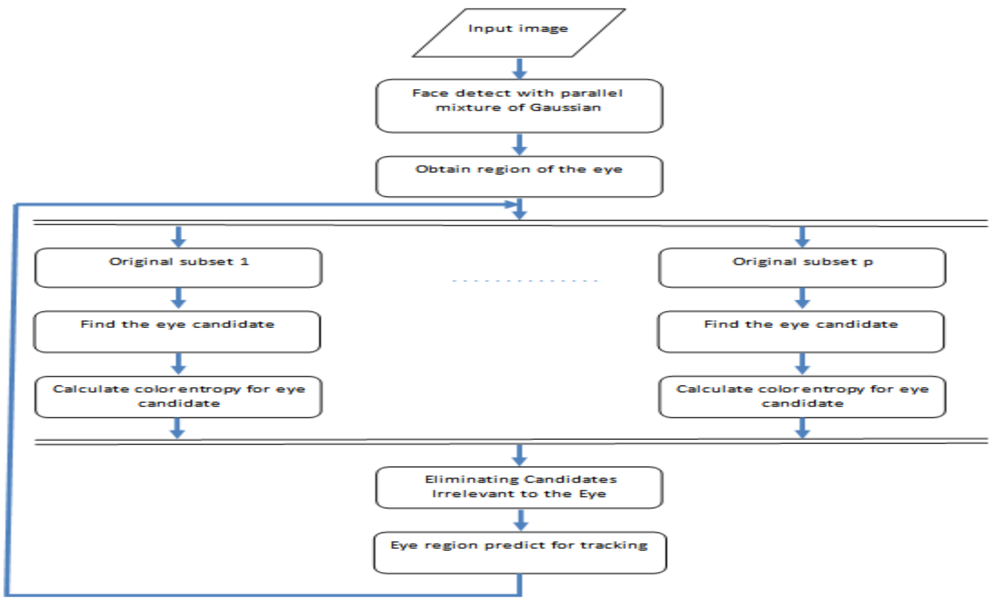


Fig. 2. Proposed algorithm for eye detection and tracking.

□□□□□□□□□□□□□□□□□□□□□□□The horizontal projection for an image of $N_{Rows} \times N_{Columns}$ is a sum of pixel intensities in each line. Equation (6)

$$vp = \{vp_x | 1 \le x \le N_{Rows}\}$$
$$vp_x = \sum_{y=1}^{N_{Columns}} f(x, y) \qquad (6)$$

Fig.3 is an image of the face along with its horizontal projection. As observed in this figure, the uppermost valley since the region above the eyebrows and below the eyes is brighter, the eye region is cropped. This region, which starts above the eyebrows and extends to the region below the eyelids, includes two eyebrows, two eyes, and their characteristic parts like eyelids and eye corners. Applying Harris method, we roughly locate the pupil. Having identified the pupil in the initial frame, we detect the eye region by Kalman filtering. Fig.2 illustrates the proposed method.

3.2. Eye detection algorithm

In this stage, eye candidates are extracted through the following procedure: First, the values of A, B, C are identified through equation (7).

$$A = (I_x)^2, B = (I_y)^2, C = (I_x I_y) \qquad (7)$$

Where I_x, I_y denote the image gradients in x, y directions, respectively.

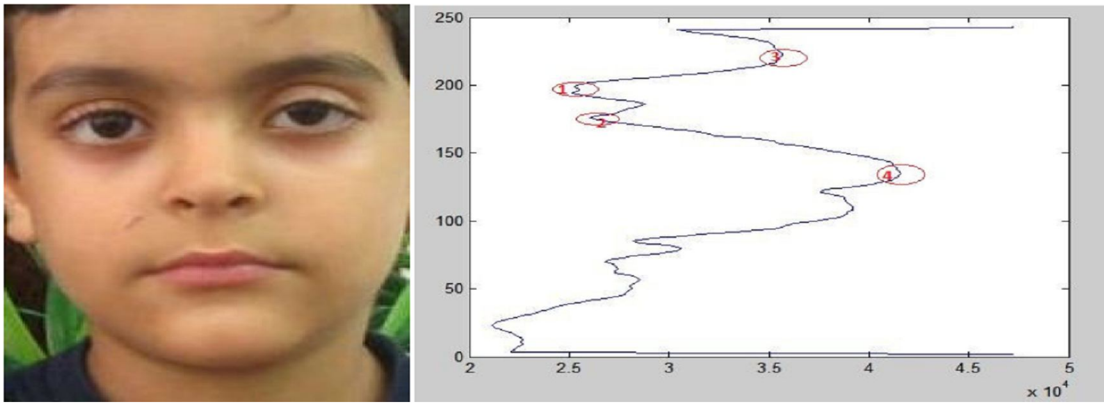


Fig. 3. Horizontal projection of face: the horizontal axis illustrates the total pixels of each line of the face and the vertical axis the number of each line

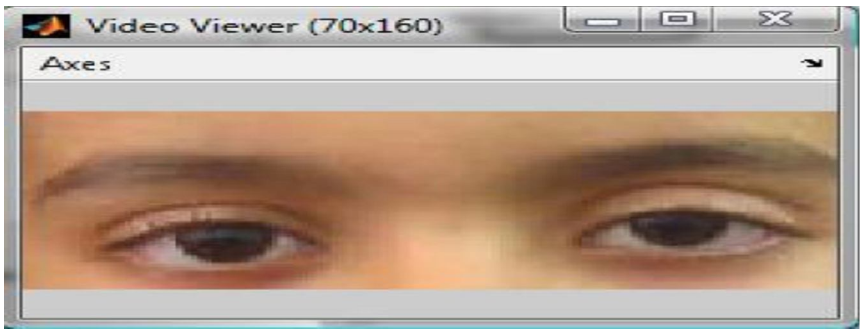


Fig. 4. Extraction of the eye region obtained from the horizontal projection of the face

Then R is obtained from equation (8):

$$R = Det(M) - KTr^2(M) \qquad (8)$$
$$Tr = \alpha + \beta = A + B, Det(M) = \alpha\beta = AB - C^2$$

The points with the highest values of R in the specified region are selected as the eye candidate's. In the next stage, the eye region is well detected among these points. Color entropy in the eye region is used to eliminate the irrelevant candidates.

3.3. Eliminating Candidates Irrelevant to the Eye

The image of eye has a lot of features. Variations in lightness levels and the form of eye component complicate the detection of these features. One factor useful for locating the eye is the irregular variations in lightness levels of sclera (eye's white area). The blood vessels in this area contribute to this entropy on the surface of the sclera. To ensure correct detection of pupil, some points with the highest values of r are specified. These points are considered as the pupil candidates. To eliminate the candidates irrelevant to the pupil, color entropy [6] is considered for each candidate, and color entropy is calculated. The probability of entropy is obtained from equation (9).

$$H(S) = -\sum_{i=1}^{n} p(x_i)\log_2^{p(x_i)} \qquad (9)$$

Where $p(x_i)$ denotes the probability of pixel x_i, and n stands for the number of pixels in each section. The probability function of each pixel, $p(x_i)$, is the frequency of the pixel color intensity in the selected area divided by the total frequency of pixels. Two regions with the highest entropy in the eye's candidate are decided as the area including the pupil. The center of these areas is regarded as the center of the pupil.fig.5.

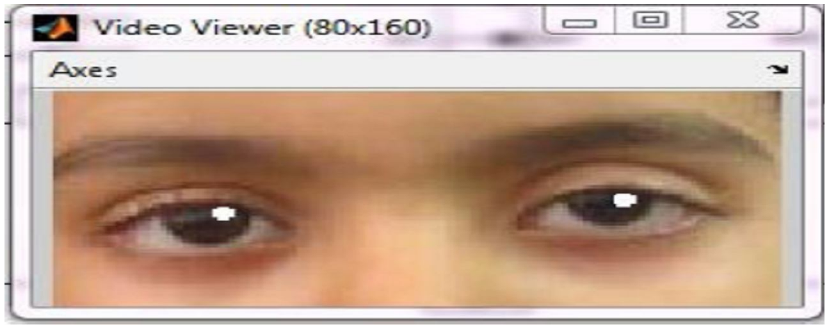


Fig. 5. Identifying the pupil by the proposed method

3.4. Eye tracking

In most application, eye tracking or face tracking follows eye detection. To cover head movement, the eye itself is not tracked. Rather, part of the face including two eyes is tracked. Among the method available, Kalman filtering is faster and computationally less intricate, and therefore will be used for detecting the eye region. Kalman filter [7] predicts the next time state vector based on the movement information in the previous stages. For detection, the state vector is considered $X = (x, y, v_x, v_y)$. x, y Are the coordinates of the center of the region including the eye, v_x, v_y the coordinate difference of the center of the eye region lies in two successive frames. Kalman filter estimates the state vector at $t+1$ from the time state vector t. Equation (10).

$$X_{t+1} = \Phi X_t \qquad (10)$$

The observation vector is supposed as $z = (x, y)$. The equation (11) relating the state vector and observation vector is:

$$Z_{t+1} = H \ X_{t+1} \qquad (11)$$

Where Φ Matrix is as follows:

$$\Phi = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

H Is the matrix that connects X, Z vectors and is defined as:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The proposed method will be put into practice in a region selected based on the estimated observation matrix (Z_{t+1}). This narrows the operational domain and requires fewer calculations. After the exact coordinates of the region including the eye have been obtained, the new state vector is derived from equation (12) using the derived state vector. Kalman filtering corrects its parameters for estimation in the following stages.

$$x_{t+1} = x_{t+1}^{-1} + k_{t+1}\left(z_{t+1} - Hx_{t+1}^{-1}\right) \qquad (12)$$

In this equation x_{t+1}^{-1}.is the estimated state vector, and k_{t+1} represents the gain matrix, which is derived from equation (13)

$$k_{t+1} = \frac{\sum_{t+1}^{-1} H^T}{H \sum_{t+1}^{-1} H^T} \qquad (13)$$

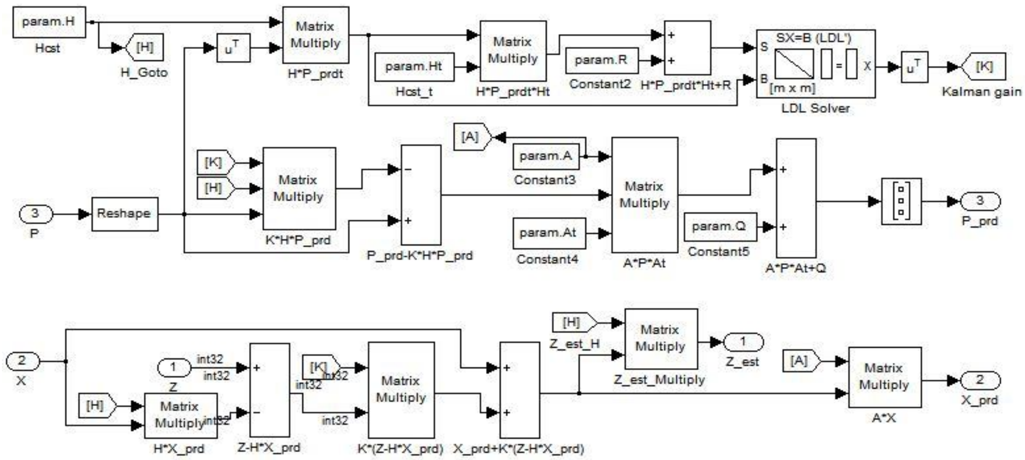
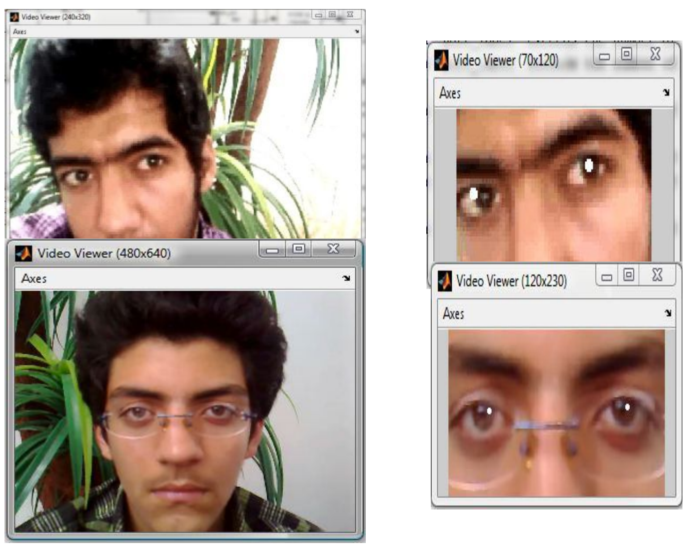


Fig. 6. Implementation of Kalman filter algorithm

4. Observation

In this section the experimental results obtained on a large number of images are shown. The images taken from Banka database. No constraints have been imposed on the background. Fig.7 shows the result of eye tracking by using Kalman filter and pupil detection in different frames. In fig.8 and fig.10 some results of correct detection are shown, in particular fig.9 shown the eye detection on image of people wearing glasses.



Fig,7. Pupil detection

Fig. 8. Result of the detection of a person wearing glasses in different frames

Fig.9. Some result with people wearing glasses

The proposed method is a step-by-step process and the results of a subsequent step necessarily depend on the previous step. Accordingly, if the first stage is not precise, the final detection will not be accurate enough, and the offered procedure will failed. Based on the derived results, it could be claimed that the method enjoys a high detection rate.

Fig.10. Some result

4. Conclusion

In this paper we have proposed a parallel algorithm for detecting eyes in color images with complex background. To locate the pupil, first the face was extracted from the whole image. Then the pupil position was found using the proposed method. Since computation of this formula on every single pixel is independent of each other, it can be parallelized as follows. The master processor is responsible for dividing the image into p portions and sends each portion to the corresponding Slave processor. Considering p processors, n/p rows of an image can be assigned to one processor. The position of the eye region in the next frame was estimated by using Kalman filter. An area surrounding the estimated location was selected and the proposed method was carried out within it. This narrowed the operational domain and required fewer calculations. Based on the obtained practical results, the proposed method could be assessed as an appropriate method to eye detect and track. Results indicated that the method could locate the position of the pupil well.

REFERENCES

1 D. Hansen Q.Ji," In the Eye of the Beholder: A Survey of Models for Eyes and Gaze", Pattern Analysis and Machine Intelligence, IEEE Transactions on March 2010

2 E. Rosten and T. Drummond. “Fusing Points and Lines for High Performance Tracking”; Proceedings of the IEEE International Conference on Computer Vision Vol. 2 (October 2005): pp. 1508–1511.Ba Linh NGUYEN, “Eye Gaze Tracking”; IEEE vol. 2,2009.

4 J. Zhang, Y. Liu and S. Ha," A Novel Approach of Face Detection Based on Skin Color Segmentation and PCA",The 9th International Conference for Young Computer Scientists, IEEE 2008

5 P. Gejgus, J. Placek, M. Sperka," Skin color segmentation method based on mixture of Gaussians and its application in Learning System for Finger Alphabet",International Conference on Computer Systems and Technologies - CompSysTech’2004

6 M.Hassaballah, and S. Ido." Eye Detection Using Intensity And Appearance information"IAPR conferecnce on machine vision application,Japan,May 2009

7 Z Zhu, Q. Ji, K. Fujimura," Combining Kalman Filtering and Mean Shift for Real Time Eye Tracking Under Active IR Illumination", IEEE 2002

8 T. D'Orazio, M. Leo, G.Cicirelli, and A. Distante, “An Algorithm for real time eye detection in face images”, in Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04),Vol.3, 2004.

9 K. Lin, J. Huang, J. Chen, and C. Zhou, “Real-time Eye Detection in Video Streams,” Fourth International Conference on Natural Computation (ICNC'08), pp.193-197, Oct., 2008.