Human Eye Tracking Using Particle Filters

Elham Ghasemi-Dehkordi[1], Mahmoud Mahlouji[2] and Hossein Ebrahimpour- Komleh[3]
[1]Department of Computer Engineering, Dezfoul Branch, Islamic Azad University, Dezfoul, Iran

[2]Department of Electrical and Computer Engineering, Kashan Branch, Islamic Azad University, Kashan, Iran

[3]Department of Computer Engineering, the University of Kashan, Kashan, Iran

Abstract
In this paper, an effective method for human eye tracking and also decreasing the current challenges and problems in its algorithms has been proposed. In this method, firstly face has been detected and segmented from the remaining parts to make the searching area, in tracking stage, narrower and processing speed higher. Then eye area is determined and eye pupils are detected in the specified area. In the proposed method, to support tracking in eyes closure state, corner detection has been additionally used. Experimental results were considered on a variety of images with existence head and face rotation, make up and eyes closure show the ability of this method in human eye tracking.
Keywords: *Eye Detection, Face Detection, Eye Tracking, Particle Filter, Corner Detection, Harris Algorithm.*

1. Introduction

Nowadays by ever-increasing development of scientific discussions in the content of computer vision and pattern recognition, automatic eye tracking in video images has been significantly considered because of eye organism and structure. Since eye is an important and critical organ of the body and manifests some internal moods and their reflexes to physical and non-physical factors, by using it some moods like fatigue, sleepiness, drunkenness, happiness, and sadness, stare, etc can be distinguished. So, design and implementation of eye tracking systems are applicable to the following cases: safety systems to control peoples entrance and exit, tracking, detecting and determining social misdemeanant identity, traffic systems to recognize drunken and narcotic-taken drivers, lie detector systems and in cars to detect driver fatigue and sleepiness. An eye tracking system includes the following stages: image acquisition; face detection; eye detection; and eye tracking.

Eye detection methods are divided into four classes [1]: shape-based methods; feature-based methods; appearance-based methods and hybrid methods. When eye is open, it has some special form. Eyes' corners, pupil and eyelids cause it to be distinguished from other parts of face. Shape-based methods use these characteristics and express some models for eye. This model can be a simple ellipsoid or a complex structure [2, 3]. Feature-based methods use some distinguishing features around the eye. Iris, pupil, eyelids, eyelashes, limbus (border between iris and sclera), pupil expression and reflection of light in cornea are some features used for detecting the eye [4, 5]. Appearance-based methods which are known as pattern matching methods or general methods detect and track the eyes directly based on their appearance states. These methods are independent of the considered objects and can model every other object beside the eye [6,7]. The main purpose of hybrid methods is to compose various advantages of eye models at a system to overcome relative constraints. To improve these algorithms, the proposed algorithms try to reduce search area for pattern matching by combining methods, or propose an approach reducing the time of this pattern matching [8,9].

The available approaches for eye tracking can be divided into three classes [10]: knowledge-based methods, learning-based methods and motion estimation methods. In knowledge-based methods, tracking methods are defined and developed based on the rules obtained by studies and researches on face components. Simply we can utilize some rules to describe face features and their relations [11,12]. Learning-based methods can be classified into the three following categories [13,14]: neural networks, Adaboost classifiers and support vector machines. Motion estimation methods estimate the object position in current frame, and then determine its exact position by a local searching. In the next stages, by a correct estimation in the first stage, the accuracy and effectiveness of the method are increased [15-17].

In this paper, an effective method is presented for eye tracking. To perform this, in eye and face detection stages, color feature has been used. Then in tracking stage, pupil position estimation method and particle filter have been used. To support eye closure in tracking stage, shape-based method is combined with pupil position estimation method and parallel to particle filter, corner detection has been used. The remainder of this paper is as follows: in the second section, and eye tracking system will be introduced. In the third section, the proposed method is presented. In the fourth section, the results obtained by experiments are offered and the fifth section is for conclusion.

2. Eye Tracking System

In general, an eye tracking system can be outlined in four stages:

- Image acquisition
- Face detection
- Eye detection
- Eye tracking

Image acquisition stage is data or image entry to system which can be a video capturing by a camera. The output of this stage is an image in RGB space. After image acquisition, face detection is performed. One current method for face detection is by using color feature [18]. As the RGB space, besides having pixel color data, has light intensity data, and also face brightness is different in various people and environments, use this space to detect face color is not effective and makes some problems. So we should find a space not dependent on light intensity. One option is YCbCr chromatic color space. Firstly the image is transformed to YCbCr. RGB color space is transformed to YCbCr as:

$$\begin{cases} Y = 0.299R - 0.587G - 0.11B \\ Cb = R - Y \\ Cr = B - Y \end{cases} \quad (1)$$

After transforming to YCbCr, the mean and covariance matrix of Cb and Cr components are computed. Eq. (2) shows how to compute the mean and covariance matrix.

$$\text{mean}: \mu = E(x),$$
$$\text{cov}: c = E\left((x-\mu)(x-\mu)^T\right) \quad (2)$$

in which E is mathematical expectation, μ is mean, c is covariance matrix, x is the main matrix and $(x-\mu)^T$ is $(x-\mu)$ transposed. By using mean and covariance measures, a Gaussian model for skin color can be determined. After obtaining skin color model, you can use Gaussian function and Mahalanobis distance for face detection and recognize a pixel belong or non-belong to the considered model.

After face detection, eye detection is performed. Since eye color differs from other parts of face, this feature can be used for eye detection. For eye detection based on color feature, we can use horizontal and vertical projection [19]. Horizontal projection for an image with M rows and N columns is obtained from adding pixels intensities in each row according to Eq. (3) and vertical projection is obtained from adding pixels intensities in each column according to Eq. (4) as:

$$hp = \{hp_x \mid 1 \le x \le M\}$$
$$hp_x = \sum_{y=1}^{N} f(x,y) \quad (3)$$

$$vp = \{vp_y \mid 1 \le y \le N\}$$
$$vp_y = \sum_{x=1}^{N} f(x,y) \quad (4)$$

Another method of eye detection is by using eye corners. In [20], Harris algorithm is used to specify the pupil's center. In this algorithm, a mask in each point of image is considered. By displacing mask in different directions the mean variation rate of image intensity in each window, compared with the main window, is computed and the minimum variation is considered as corner response. Depending on the position of each point, three states are created:

- If the windowed image patch is flat (approximately constant in intensity), then all shifts will result in only a small change.
- If the window straddles an edge, then a shift along the edge will result in a small change, but a shift perpendicular to the edge will result in a large change.
- If the windowed patch is a corner or isolated point, then all shifts will result in a large change. A corner can thus be detected by finding where the minimum change produced by any of the shifts is large.

Harris algorithm is mathematically expressed as:

$$E_{x,y} = \sum_{u,v} w_{u,v} \mid I_{x+u,y+v} - I_{u,v} \mid^2 \quad (5)$$

in which w is a mask imposed on image, which is considered as circle mask with coefficients 1. I is the image gradient and E is the variation made by (x, y) displacement. (x, y) displacement in four directions includes the following set:
{(1 ,0),(1,1),(0, 1),(0,0)}

Harris corner detector introduces local maximum of $\min_{x,y}\{E_{x,y}\}$ that is more than a special threshold, as a corner. Equation (6) includes all possible little displacement as:

$$E(x,y) = Ax^2 + 2Cxy + By^2$$
$$A = (I_x)^2 \otimes w,$$
$$B = (I_y)^2 \otimes w,$$
$$C = (I_x I_y) \otimes w \quad (6)$$

in which, w is Gaussian mask, I_x, I_y is image gradient in is convolution $\otimes$x and y directions, respectively, and operator. E can be expressed as:

$$E(x,y) = (x,y)M(x,y)^T,$$
$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \quad (7)$$

in which, M is a 2*2 matrix in (x, y) coordinates and E is local auto–correlation function.

In [20], it has been shown that corner response, R, can be obtained in each image point as:

$$R = \mathrm{Det}(M) - K\mathrm{Tr}^2(M)$$
$$\mathrm{Det}(M) = \alpha\beta = AB - C^2 \quad (8)$$

in which K is constant. The corner response in corner area is positive, in edge area is negative and in flat area, has a minimal absolute magnitude. To reduce the computation time, the existing mask in Harris algorithm is neglected. So, firstly A, B, C are obtained as:

$$A = (I_x)^2$$
$$B = (I_y)^2$$
$$C = (I_x I_y) \quad (9)$$

After computing these measures, R is obtained by Eq. (8). Experiments show two points with the highest value of R in an eye area are eye iris corners [20].
The last stage in eye tracking system is eye tracking stage. One most favored and applicable method useful for real time applications is particle filter [21,22].
Particle filter is a method numerical and computational to obtain the probability density function of a random process and the estimation of a specific parameter from on it and in the event are an extended chain Monte Carlo methods [21,22], since the predominant method has been using Kalman filter or extended Kalman filter. These methods usually assume for the case when the model is linear and Gaussian noise, to obtain the optimum solution problem. But in other cases, depending on the remoteness of these assumptions are far from optimum. In other words, since the particle filter is a numerical method for estimating signal, nonlinear model or non-Gaussian noise for it does not matter. In such cases, the particle filters with high performance in various fields such as communication, signal processing, tracking and many other fields are used.
The idea using of particle filters for tracking an object in a sequence of images initially were introduced separated by a few research groups on the field and among those who were working in the fields of computer vision and image processing are known to be dense algorithm. This method mainly is based on the edge detection methods; another powerful method for target tracking in particle filters is using color distribution model target or histogram model for target tracking.
Color histogram of an object creates for its tracking many benefits. Specifically, in particle filters a target is tracked based on comparison its color histogram with histogram of samples location and using Bhattacharyya distance.
Color distribution $p_y = \{p_y(u)\}_{u=1,2,\ldots,m}$ where y is calculated within the area around the target.

$$p_y^{(u)} = f\sum_{i=1}^{I} K\left(\frac{\| y - x_i \|}{a}\right)\sigma[h(x_i) - u] \quad (10)$$

Where u is discrete element in color and m is the number of elements, K is a function of the weight that to samples farther away of the target gives lower weights,

H_x and H_y respectively are the length of half axes the horizontal and vertical in the area around the target, $a = \sqrt{H_x^2 + H_y^2}$ used to adjust the size of the area and f is a normalization factor that ensures that the following relation is established.

$$\sum_{u=1}^{m} P_y^{(u)} = 1 \quad (11)$$

In the tracking stage, the state estimated at every time step, based on the new observation function is updated. So a function of observation or measurement that is based on color distribution is required. A popular measure between two distributions p(u) and q(u) is the Bhattacharyya coefficient.

$$\rho[p,q] = \sum_{u=1}^{m} \sqrt{p^u q^u} \quad (12)$$

What the higher is this coefficient; the two distributions are similar together, so that for two distributions, normalized coincident is $\rho = 1$, which is expression of full implementation of two distributions. The distance between two distributions is defined as follows:

$$d = \sqrt{1 - \rho[p,q]} \quad (13)$$

Color based tracking algorithm for updating calculated the previous density function by particle filter is used from Bhattacharyya distance. Each sample of density function represents the area around the target that is given as follows:

$$S = \left[x, x^{\bullet}, y, y^{\bullet}, H_x, H_y, \theta\right]^T \quad (14)$$

Where x and y specify the center of the area around the target, $x^{\bullet}$ and $y^{\bullet}$ represent the speed, H_x and H_y are the length of half axes in the area around the target and θ is the rotation angle of the ellipse.
Dynamic model or state of model that represents the samples changes over time is defined as follows:

$$S_k = AS_{k-1} + W_{k-1} \quad (15)$$

Where A for motion model with constant acceleration is as follows:

$$A = \begin{bmatrix} 1 & \Delta & \circ & \circ & \circ & \circ & \circ \\ \circ & 1 & \circ & \circ & \circ & \circ & \circ \\ \circ & \circ & 1 & \Delta & \circ & \circ & \circ \\ \circ & \circ & \circ & 1 & \circ & \circ & \circ \\ \circ & \circ & \circ & \circ & 1 & \circ & \circ \\ \circ & \circ & \circ & \circ & \circ & 1 & \circ \\ \circ & \circ & \circ & \circ & \circ & \circ & 1 \end{bmatrix} \quad (16)$$

W is the vector of Gaussian white noise whose covariance matrix is calculated by the following relation:

$$R = \begin{bmatrix} Ry & O_{4\times 3} \\ O_{3\times 4} & Re \end{bmatrix} \quad (17)$$

Where Ry is covariance matrix of the position and the speed of target and computed as follows:

$$Ry = \sigma_y \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & \circ & \circ \\ \frac{\Delta^2}{2} & \Delta & \circ & \circ \\ \circ & \circ & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ \circ & \circ & \frac{\Delta^2}{2} & \Delta \end{bmatrix} \quad (18)$$

And covariance matrix of the area around the target, Re, is as follows:

$$Re = \begin{bmatrix} \sigma^2_{H(x)} & \circ & \circ \\ \circ & \sigma^2_{H(x)} & \circ \\ \circ & \circ & \sigma^2_{\theta} \end{bmatrix} \quad (19)$$

To weight the samples, Bhattacharyya coefficient is calculated between the histogram of the target and the histogram of the area that each sample shows. The area of each sample is determined with its state vector, $S^{(n)}$, and since samples that their color distribution is the most similar color distribution of the target are required, more weight should be given to them with less Bhattacharyya distance. The weight of the samples, shown here with $\Pi^{(n)}$, is defined as follows:

$$\Pi^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(1-\rho[p,q])}{2\sigma^2}} \quad (20)$$

Therefore weights of samples are calculated from a Gaussian distribution with the variance, σ^2 as how to select the appropriate number for σ depends on the desired tracking problem. As a result of this weighting, samples are that more compatible with target, greater probability have that in several iterations of algorithm remain and samples with less compatible, greater probability to be removed.

Finally, the estimation of the target state vector is obtained as follows:

$$E[S_k] = \sum_{n=1}^{N} \Pi_k^{(n)} S_k^{(n)} \quad (21)$$

Where N is the number of samples which is selected based on the accuracy required and volume of calculations which can be performed.

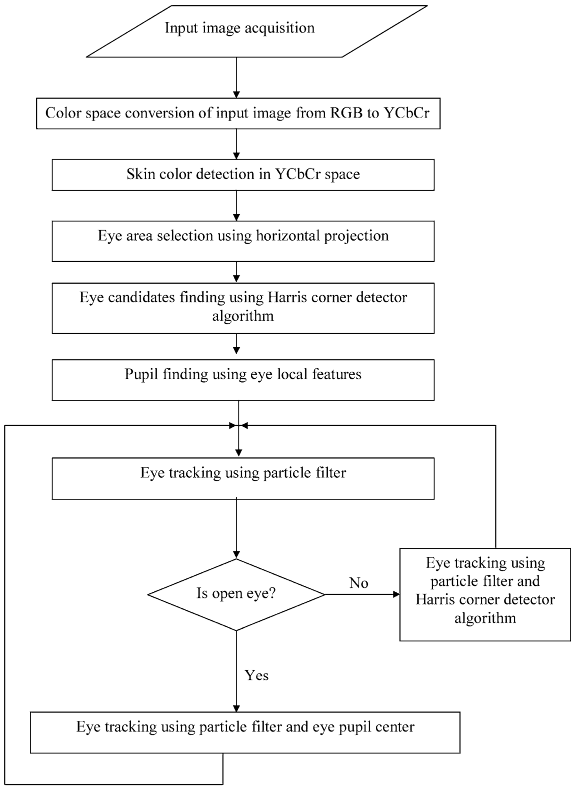


Fig.1 Proposed method algorithm for eye tracking

3. The Proposed Method

At the first, to make the intensity around ineffective in the proposed algorithm, the color space of input image is transformed from RGB to YCbCr by Eq. (1). Then a color model, according to Eq. (2) is defined for face and finally, the face area based on Gaussian model and Mahalanobis distance and using Eq. (3) is detected.

In the next stage, by using horizontal projection according to Eq. (3), the region including eyes is determined and divided into right and left areas, to precisely find the pupils' center and in each segmented area; the eye pupil's center and in each segmented area; the eye pupil's center is searched. This operation causes the searching area to be narrower. After that, by using vertical and horizontal projection via Eq. (3) and Eq. (4) the pupils' center is obtained. In the last stage, two particle filters track the eyes in parallel. Also, corners detections are used for eye tracking.

The proposed method includes four stages: face detection, detection of area including eyes, obtaining the eyes' pupils and corner detection, and eye tracking.

3.1 Face Detection

Since color is fixed when the head rotates and its processing is faster than other features of face, this feature is used for face detection. The most current color space is RGB which has many applications and is the standard space to present color images. But because of what is explained in section 2, the image is transformed into YCbCr space by Eq. (1). Fig.2 shows the differences between a person's images in color space RGB and YCbCr, Cb component and Cr component.

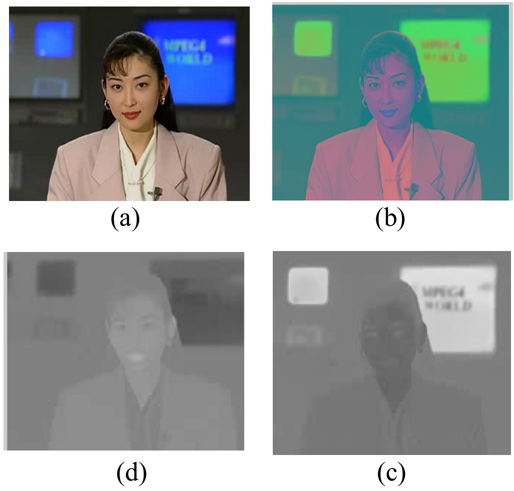


Fig.2 Differences between a person's images in various color spaces: (a) RGB space (b) YCbCr space (c) Cb component (d) Cr component

3.1.1 Finding Skin Color Model

To create skin color model, firstly a set of various skin images with different color and texture is prepared, then is transformed into YCbCr color space. After that by using Eq. (2), skin color model is obtained. In Fig 3 there are some samples of skin color.

3.1.2 Mahalanobis distance

After obtaining skin color model, by using a Gaussian model and Mahalanobis distance, every pixel's belong or non-belong to the model is determined. Figure (4) shows the result obtained by face detection.

Fig.3 Some samples of skin color

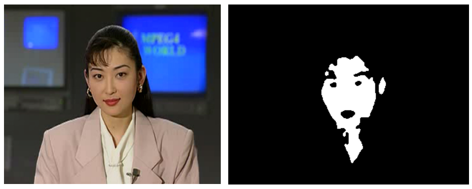


Fig.4 Result obtained by skin color detection

One of difficulties that there is in determining the skin color areas, similarity pixels color of image is with skin color model that belongs to any areas we are considering that this increases the computational complexity and in the proper functioning of the system occurs interference. For this propose, the largest area under skin color is selected as the original area, to do this work easily by calculating the area of each connection zone is possible. The region with largest area is considered as face and extracted from the image. When we have some region with the areas almost identical, the ratio of length of major to minor axis is used. In Fig. 5 can be seen the result of this process.

Fig.5 Result of face detection

3.2 Eye area extraction

After the face region detection, will turn to eye area extraction. This area includes two eyes, two eyebrows and distinctive features such as eyelids and the corners of the eyes. This area can be extracted from the horizontal projection of face area that can be calculated from Eq. (3). In Fig.6, horizontal projection of a face is observed.

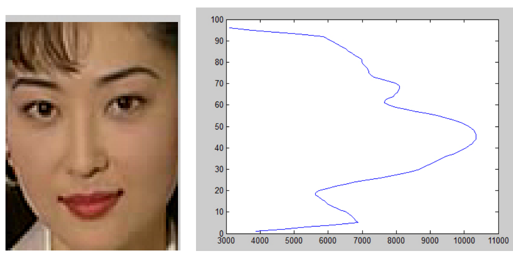


Fig.6 Horizontal projection of the face

As shown in Fig. 6, the first valley from above is related to place of pupil. Because of drastic changes in the eye area, the valley cannot easily be extracted. According to Fig. 6, because the forehead and below the eyes are brighter than pupil, instead of considering the minimum point on the graph, two local maximum point on the graph can be used to obtain the eye area.

From above face to down, it can be observed that two values of maximum horizontal projection, respectively, the one belonging to the forehead area and the other belongs to the area between eyes and mouth (the septum). According to the physiology of the face, this observation seems to be true because the flat area of the face, having the brightness level relatively high, is located in these regions. Figure 7 shows the result obtained from eye area extraction of face area.

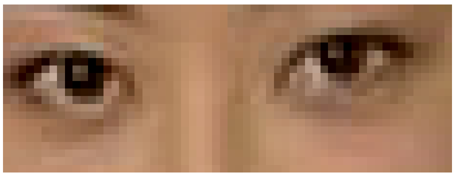


Fig.7 Eye area detection using horizontal projection

3.3 Finding Eye Candidates

To find eye candidates within the area obtained in the previous step is used Harris algorithm. Firstly, the image gradient is calculated and then, values of A, B, C from Eq. (9). By substituting A, B, C in Eq. (8), the value R can be calculated for each pixel of image. Points with the highest value of R can be introduced as candidates of iris. To remove unrelated candidates with iris, feature of color entropy in the area around iris is used. The area in size 50*50 pixels is considered for each of the candidates and entropy of color is calculated as follows:

$$H(S) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \qquad (22)$$

Where $p(x_i)$ and n are the probability of pixel x_i and the number of pixels per unit, respectively. Probability function, $p(x_i)$, for each pixel, is produced from counting the number of times that the pixel color intensity is repeated at selected portion to the total number of pixels. From candidates of eye, selected by Harris algorithm, two points are selected with the highest level of color entropy. Figure 8 shows that eyes have the highest degree of color entropy, as compared to the rest of the face components.

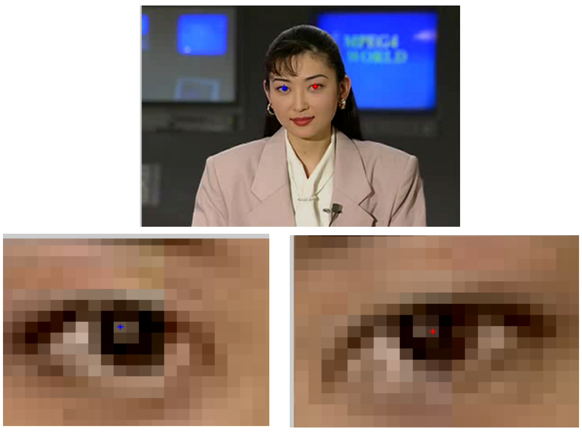


Fig.8 Degree of color entropy in Eye areas – from candidates selected by Harris algorithm, eyes that have the highest degree of color entropy are shown in the second row of images.

In Fig. 9, samples from detected corners of eyes are observed

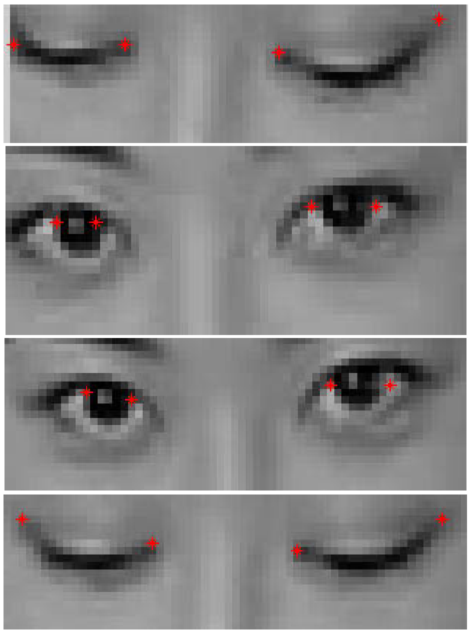


Fig.9 detected Corners of eyes

In Fig. 10, samples of pupil detection are observed.

Fig.10 Pupil detection

3.4 Finding the Pupil Center

By having a pixel from iris or pupil and having pupil circle, pupil center can be obtained. To find pupil center, lines intersection can be used. In this method, only one row and column of eye is processed instead of total eye area. So, a line is drown, along with x and y axis to the border edge of iris and sclera. As shown in Fig.11, if the intersection center of drown lines is considered as (x, y) and the end points of lines are considered as $(x_1, y'), (x_2, y'), (x', y_1), (x', y_2)$, the pupil center can be obtained as:

$$x_c = \frac{x_1 + x_2}{2}, \quad y_c = \frac{y_1 + y_2}{2} \qquad (23)$$

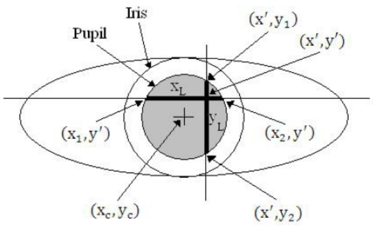


Fig.11 Using lines intersection to find pupil center

3.5 Eye tracking

In most applications, after eye detection, tracking of eye or face movements is considered. The same problems in eye detection state are in eye tracking, by a difference that in the tracking cases, the problems appears stronger and there is more destructive effect on tracking process. As an example, changing the environment light conditions cause loss of eye position on the image and consequently eye detection process is repeated.

For eye tracking, in addition to the estimation and location of eye positions in sequential frames, head and face movements should be considered and a method not sensitive to these changes and problems should be proposed. In addition to head, face components like eyelids have independent movement which should be considered.

To support head movements, the eye itself was not tracked and an area containing eye was tracked.

Among proposed methods, is used from particle filter. In the proposed method, two filters used in parallel, track the eyes separately because as explained in section 3-3, the eyes are separately recognized. In Fig. 12, results of eye tracking using particle filter and pupil detection in the different frames can be observed.

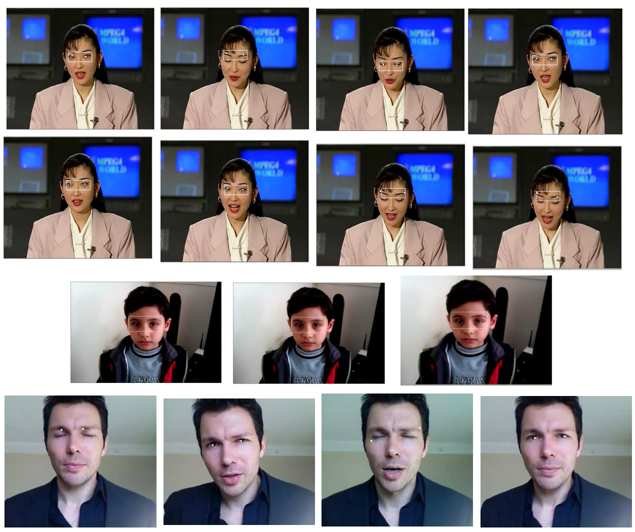


Fig.12 Eye tracking using particle filter

A problem here is that if both eyes are closed, Particle filter does not work. To overcome this problem, besides particle filter, the eye corners features are also used. To perform this, in each left and right area, two corners are recognized. To find the corners, Harris algorithm, explained in section 2 has been used. So, the proposed method for tracking, is a combination of particle filter and corner detection.

In proposed method, this problem is considered that in particle filter for each of the particles there is probability degree that much particles are farther from the area around the target, this probability is lower number; the average of probabilities of all particles are calculated when the eye is open, the average probabilities of particles is a high number but when the eyes are closed, the average probabilities of particles is a low number that this probability degree of a particle indicates the degree of similarity of each particle to the reference histogram. Threshold value is considered experimental. If in working frame, the average probabilities of the particles were higher than the threshold, eye is open and eye tracking does not fail in this case using particle filter. But if the average probabilities of particles were lower than the threshold, eye is close and in this case, eye tracking fail using particle filter. To overcome this problem, Harris corner detection algorithm is used and as mentioned before, corners of eyes are detected and used in eye close state for eye tracking.

Although in this method, the required hardware has been increased and for each eye, an additional filter has been used, but tracking with head and face rotation existence, make up existence and eyes closure is performed precisely.

4. Experimental results

Fig.13 shows the results of eye tracking and pupil detection using proposed method in different frames. In this figure, it can be observed that eye closure problem is resolved and the eye in closing state is also tracked. Fig. 14 and 15 are supported the results of eye tracking and pupil detection using proposed method in various states of head and face movements existence, make up existence and eye closure.

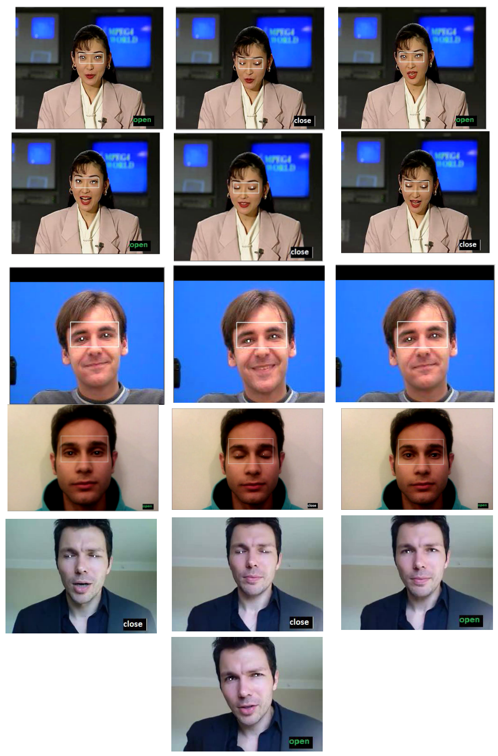
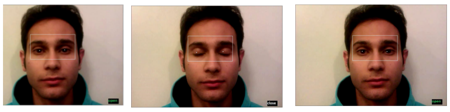
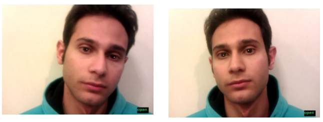


Fig.13 Eye tracking using proposed method

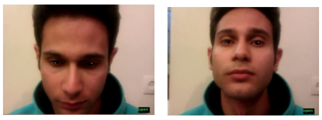
Fig.14 Results of the proposed method for eye tracking with make up



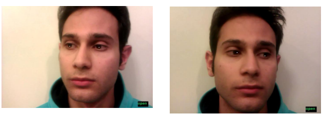
(a)



(b)



(c)



(d)

Fig.15 results of eye tracking in different frames using proposed method in:
a) Normal state b) Head rotation state horizontally
c) Head rotation into high and low d) Head rotation state vertically

Comparison between the proposed method and other ones is difficult because of their performing in different platforms and databases. So, the main parts of these methods are theoretically compared. In Table 1, the proposed method is compared with two similar methods.

Table 1: Comparison between the proposed and two different methods

	Used Method for detection and tracking	Detection in different light conditions, head rotation, make up and eye closure	Image with high resolution
Paper[5]	Entropy of eye and darkness of iris	Supports the different light conditions , head rotation	Need
Paper[12]	Eye form structure and corners	Supports head rotation	Need
Proposed Method	Combine color intensity around the eye, particle filter and corner detection	Supports head rotation to 35 degree, different light conditions, make up and eyes closure	No Need

Following results are concluded from Table1:
In [5], entropy of eye and darkness of iris is used. This method supports various light conditions, head rotation and in case of existing make up and eyes closure tracking is not possible. Also the images used by this method should be in high resolution.
In [12], the eye form, structure and corners are used for detection. This method only supports head rotation and in the case of different light conditions, make up and eyes closure, it cannot track. Also, the images used by this method should have high resolution.
In the proposed method, the combined color intensity around the eye, particle filter and corner detection are used for detection and tracking. The proposed method supports various light conditions, head rotation, make up and eye closure. Also, in this method, there is no need to the high resolution images.

5. Conclusions

In this paper, a new method for human eye tracking using particle filter has been presented. Considering the experiments in different conditions including images changing light conditions, head rotation, make up and eye closure, good results were obtained. Also, experimental results show the proposed method has low computational complexity and maximum stability.

References

[1] S. Yamamoto and V. G.Moshnyaga, “Algorithm Optimizations for Low-Complexity Eye Tracking”, *IEEE International Conference on Systems, Man, and Cybernetics,pp.18-22,* 2009.
[2] K.N. Kim and R.S. Ramakrishna, “Vision-Based Eye-Gaze Tracking for Human Computer Interface”, *IEEE International Conference on Systems, Man, and Cybernetics,* Vol. 2, pp. 324-329, 1999.
[3] R. Kothari and J.L. Mitchell, “Detection of Eye Locations in Unconstrained Visual Images”, *IEEE International Conference on Image Processing,* Vol. 3, pp. 519-522, 1996.
[4] W. Liting, D. Xiaoqing and F. Chi, “Face Live Detection Method Based on Physiological Motion Analysis”, *Tsinghua Science & Technology,* Vol. 14 , Iss. 6, pp. 685-690, 2009.
[5] M. Hassaballah and S. Ido, “Eye Detection Using Intensity and Appearance Information”, *IAPR Conference on Machine Vision Applications,* pp. 346-349, 2009.
[6] Z. Zhu, K. Fujimura and Q. Ji. “Real-Time Eye Detection and Tracking under Various Light Conditions”, *Eye Tracking Research & Applications Symposium,* pp. 139-144, 2002.
[7] A. Hyrskykari,“Utilizing Eye Movements: Overcoming Inaccuracy while Tracking the Focus of Attention during Reading”, *Computers in Human Behavior,* Vol. 22, Iss. 4, pp. 657-671, 2006.
[8] K. Peng, L. Chen, S. Ruan and G. Kukharev, “A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles”, Journal of Computer Science & Technology, Vol. 5, No. 3, 2005.
[9] K.C.K. Lee, J. Schiffman, B. Zheng, W.C. Lee and H.V. Leong, “A System for Tracking Nearest Surrenders in Moving Object Environments”, *IEEE international Conference on Pervasive Services,* Lyon, 2006.

[10] Z. Qian and D. Xu, "Automatic Eye Detection Using Intensity Filtering and K-means Clustering", *Pattern Recognition Letters,* Vol. 31, Iss. 12, pp. 1633-1640, 2010.
[11] B. Kroon, S. Maas, S. Boughorbel and A. Hanjalic, "Eye Localization in Low and Standard Definition Content with Application to Face Matching", *Computer Vision and Image Understanding,* Vol. 113, Iss. 8, pp. 921-933, 2009.
[12] Y. Tian, T. Kanade and J.F. Cohn, "Dual State Parametric Eye Tracking", *IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 110-115, 2000.
[13] F.H.C. Tivive and A. Bouzerdoum, "A Fast Neural-Based Eye Detection System", *iEEE International Symposium on Intelligent Signal Processing and Communication Systems,* pp. 641-644, 2005.
[14] Z. Niu, S. Shan, S. Yan, X. Chen and W. Gao, "2D Cascaded Adaboost for Eye Localization", *IEEE International Conference on Pattern Recognition,* Vol. 2, pp. 1216-1219, 2006.
[15] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 25, Iss. 5, pp. 564-577, 2003.
[16]Z. Zhu, Q. Ji, K. Fujimura and k. Lee "Combining Kalman Filtering and Mean Shift for Real Time Eye Tracking under Active IR Illumination", *IEEE International Conference on Pattern Recognition,* Vol.4, 2002.
[17] G. Veneri, P. Federighi, F. Rosini, A. Federico and A. Rufa, "Influences of Data Filtering on Human - Computer Interaction by Gaze-Contingent Display and Eye-Tracking Applications", *Computers in Human Behavior,* Vol. 26, Iss. 6, pp. 1555-1563, 2010.
[18] X. Long, O.K. Tonguz and A. Kiderman, "A High Speed Eye Tracking System with Robust Pupil Center Estimation Algorithm", *IEEE International Conference Engineering in Medicine and Biology Society,* pp. 3331-3334, 2007.
[19] Z. Zhu and Q. Ji, "Eye and Gaze Tracking for Interactive Graphic Display", *Machine Vision and Applications,* Vol. 15, Iss. 3, pp. 139-148, 2004.
[20] T. Liu, S. Zhu, "Eyes Detection and Tracking Based on Entropy in Particle Filter", *International Conference on Control and Automation,* Vol. 2, pp. 1002-1007, 2005.
[21] K. Nummiaro, E. Koller-Meier and L.Van Gool, "An Adaptive Color-Based Particle Filter", Image and Vision Computing, Vol. 21, Iss. 1, pp. 99-110, 2003.
[22]Y. Kuo, J. Lee and S. Kao, "Eye Tracking in Visible Environment", Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 114-117, 2009.

Elham Ghasemi-Dehkordi received the B.S. degree in computer engineering from Mobarakeh Branch, Islamic Azad University, Mobarakeh, Iran, in 2009. He is currently a graduate student in the Department of Computer Engineering at Dezfoul Branch, Islamic Dezfoul University, Dezfoul, Iran His research interests include image processing and pattern recognition.

Mahmoud Mahlouji received the B.S. degree in telecommunications engineering from Sharif University of Technology, Tehran, Iran, in 1990, the M.Sc. degree in electronics engineering from Sharif University of Technology, Tehran, Iran, in 1993, and the Ph.D. degree in telecommunications engineering from Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2008. At present he is an assistant professor of the Electrical and Computer Engineering Department, Kashan Branch, Islamic Azad University, Kashan, Iran. His current interests are in image processing, pattern recognition, neural networks, computer vision, and iris recognition.

Hossein Ebrahimpour-Komleh is currently an assistant professor at the Department of Electrical and Computer Engineering at the University of Kashan, Kashan, Iran. His main area of research includes Computer Vision, Image Processing, Pattern Recognition, Biometrics, and Robotics, Fractals, chaos theory and applications of Artificial Intelligence in Engineering. He received his Ph.D. degree in Computer engineering from Queensland University of technology, Brisbane, Australia in 2006. His Ph.D. research work is on the Fractal Techniques for face recognition. From 2005 to 2007 and prior to joining the University of Kashan, he was working as a Post-doc researcher in the University of Newcastle, NSW, Australia and as a visiting scientist in CSRIO Sydney. Hossein Ebrahimpour-Komleh has B.Sc. and M.Sc. degrees both in computer engineering from Isfahan University of Technology (Isfahan, Iran) and Amirkabir University of Technology (Tehran, Iran.) respectively. He has served as the editorial board member and reviewer of several journals and international and national conferences.