Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

Vol. 3, No. 1, March, 2022

Eye Tracking Algorithm Based on Multi Model Kalman Filter

S. H. Ziafati Bagherzadeh [1*], S. Toosizadeh [1]

[1] *Department of Electrical Engineering, School of Engineering, Islamic Azad University, Mashhad Branch, Mashhad, Iran.*

Received 14 November 2021; Revised 12 January 2022; Accepted 23 January 2022; Published 01 March 2022

Abstract

One of the most important pieces of Human Machine Interface (HMI) equipment is an eye tracking system that is used for many different applications. This paper aims to present an algorithm in order to improve the efficiency of eye tracking in the image by means of a multi-model Kalman filter. In the classical Kalman filter, one model is used for estimation of the object, but in the multi-model Kalman filter, several models are used for estimating the object. The important features of the multiple-model Kalman filter are improving the efficiency and reducing its estimating errors relative to the classical Kalman filter. The proposed algorithm consists of two parts. The first step is recognizing the initial position of the eye, and Support Vector Machine (SVM) has been used in this part. In the second part, the position of the eye is predicted in the next frame by using a multi-model Kalman filter, which applies constant speed and acceleration models based on the normal human eye.

Keywords: Eye Tracking; Multi-Model Kalman Filter; Support Vector Machine; Image Processing.

1. Introduction

Eyes and their movements are very important in the detection of disease, needs, and emotions. In other words, by considering the geometry of the eye, its movement, and state, we can figure out the needs and emotions. The necessity of studying and developing eye tracking has special importance amongst researchers of different fields. When it comes to an accident, the eye's first feature is being less vulnerable in comparison to other organs. Currently, researchers have presented various methods for eye tracking that can be used for developing useful and accurate eye tracking systems. Eye detection and tracking are used by researchers for face detection, expressing one's emotions, and face recognition. Eye tracking methods can be divided into two categories as follows:

- Passive methods based on images and active methods based on infrared light;
- Classic passive methods detect faces based on eye shape and light intensity distribution. In infrared based methods, eye detection and tracking are done by taking the eye's (pupil's) reflection of infrared rays into measure.

Classic methods in face detection can be divided into three categories including:

- Pattern-based methods,
- Appearance-based methods, and
- Specifications-based methods.

* Corresponding author: hassan.ziafati@mshdiau.ac.ir

http://dx.doi.org/10.28991/HIJ-2022-03-01-02

In the pattern-based method, based on its shape, a general model of the eye is chosen, and then the image is scanned to find the eye by pattern research algorithms. Xiong et al. [1] suggested a method based on Houph Transformation for accurate face detection: Eye tracking by using the distance between the eyes Deformable patterns are widely used for eye detection [2, 3]. In these methods, a model of the eye would be designed to match the eye's figure, which includes desirable features such as translatability, rotatability, and deformability; then, the eye would be detected by minimizing the energy. Appearance-based methods detect the eye by photometric appearance characteristics. These methods usually require a wide range of training data, eye descriptions in different positions, different facial expressions, and light conditions. Data in appearance-based methods is used to train neural networks or Support Vector Machine (SVM). Obviously, detection is based on categorization in these methods. Shah et al. (2013) [4] have used special face techniques for eye detection.

In Huang et al. (2000) [5], the eye's feature space is used for eye detection. This method benefits from desirable speed but lacks accuracy. In Huang et al. (1999) [6], image wavelet transformation and eye detection have been used with a recurrent neural network. Today, researchers use other neural network-based methods for face detection. For example, in [7], there are some proposed methods for improving eye detection by neural networks. The main weakness of appearance-based methods is the need for a large dataset in the training phase for face detection. Specification-based methods use some features like edges, color distribution, and so on for eye detection. A specifications-based method is used by Wang et al. (2016) [8]. For Vrânceanu et al. (2015) [9], and Dong et al. (2015) [10], specifications of the corners of eyes are used for face detection.

In these references, images have been used for face detection. The main weaknesses of this method are inefficiency in the presence of the hair in the front part of the eye and face direction. In Morcego et al. (2016) [11], a method is proposed for estimating the parameters of the eye and tracking it. In this method, the eye model should be designed first. The corner of the eye is traced by the Kalman algorithm. The main weakness of this method is the need for a high-quality image for eye tracking. In summary, the classic detection and tracking methods of detecting and tracking by searching for form, appearance, and eye specifications are still applicable.

In Lescroart et al. (2016) [12], and Amiel et al. (2015) [13], a wavelet filter has been used to reduce the transparency effect for eye detection. However, this filter lacks suitable efficiency for all transparencies. Therefore, classic eye detection methods have some challenges, such as face direction and the transparency of the image. In Zhu et al. (2007) [14], two new methods have been employed for face tracking based on the eye model. The first method detects the stared eye as 3D. Cornea is considered as a convex mirror in this method. Based on the features of an assumed convex mirror, the eye direction is obtained. This method is very complicated and is not feasible in practical use. In Yoo et al. (2005) [15], a method for eye tracking is presented based on pupillary light direction. Face tracking using the Kalman filter is another method for detecting eyes.

In Zhu et al. (2002) [16], a combination of Kalman filter and data classification is used for eye tracking. This is a very suitable method for face tracking. A neutral Kalman filter has been used for face detection by Zhang (2010) [17]. This paper is set to be used for detecting and estimating one's eye's position in frames of a film by using a multi-model Kalman filter algorithm. In the presented method, fixed velocity and fixed acceleration models of the multi-model Kalman filter are employed due to the dynamics of the target (eye) [18-20]. In the second section, pupillary tracking based on the Kalman filter and its equations is explained. In the next section, the simulation method and its outcome are argued. In the next section, the simulation and its results are presented. The final part of this paper provides the results and insights for more research in the future.

2. Materials and Methods

2.1. Eye Tracking by the use of Kalman Filter

Kalman filter is a set of recursive equations that are used to estimate an object's position and the degree of uncertainty in the next frame. In the Kalman filter, estimation of the next position is done by measuring data in previous frames, and then, in the next step, the previous position vector is estimated by the data of the current frame. It should be noted that the Kalman filter has a minimum Euclidean norm for estimating the object position with linear dynamics. Then, the pupil tracking is explained by the Kalman filter algorithm. In this section, consider the dynamic of eye movement as Equation 1. In this equation, x_k is the mode vector of the system dynamic in frame K, y_k is the output vector in frame K, w_k is the process noise in frame K and v_k is measurement noise in frame K. In the Kalman filter, the process and measurement noises are considered white noise.

Covariance matrices of process and measurement of noises are obtained based on w_k and v_k data. In this section, it is assumed that the signals of w_k and v_k are independent and their covariance matrix is Q and R. Also, system's dynamic mode vector is (2). In this equation, $[c_k \quad r_k]$ is eye pixel position in k frame and $[\dot{c}_k \quad \dot{r}_k]$ is eye speed in frame K. It is necessary to note that in Kalman filter, the entry is $[c_k \quad r_k]$ and the output is $[c_k \quad r_k \quad \dot{c}_k \quad \dot{r}_k]$ vector estimation. The matrix can be written based on the data.

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{1}$$

$$x(t) = [c_k \quad r_k \quad \dot{c}_k \quad \dot{r}_k] \tag{2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3}$$

Let's take $\hat{x}_{k|k}$ as posterior estimate x_k in K frame, $\hat{x}_{k|k-1}$ as prior estimate x_k in frame K, $P_{k|k}$ as posterior covariance matrix in K frame, $P_{k|k-1}$ as prior covariance matrix in K frame and K_k as the gain of Kalman filter in K frame**.** In this case, updating Kalman filter can be done in the K frame based on the Equations 4, and 5:

$$\begin{aligned} &\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} \\ &P_{k|k-1} = AP_{k-1|k-1}A^T + Q \end{aligned} \tag{4}$$

$$\begin{aligned} &S_k = CP_{k|k-1}C^T + R \\ &K_k = P_{k|k-1}H^T S_k^{-1} \\ &\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k\left(y_k - C\hat{x}_{k|k-1}\right) \\ &P_{k|k} = (I - K_k H_k)P_{k|k-1} \end{aligned} \tag{5}$$

The purpose of this section is to explain the proposed method based on multi-model algorithm for optimizing the estimation of eye position. Therefore, the block diagram of the proposed method has been studied for eye tracking in a film. Third section deals with defect detection method in a frame. Support vector machine has been used in this method. Forth section explains multi-model Kalman filter algorithm and how to use it in eye tracking.

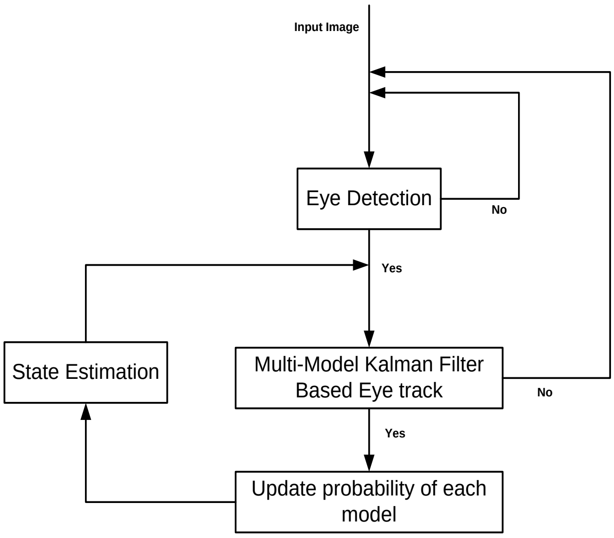


Figure 1. Block diagram of the proposed method of eye tracking

Diagram block of eye tracking algorithm is shown in Figure 1. According to this figure, in the first step, the algorithm of eye tracking can detect the first position of eye in the first frame. Clearly, support vector machine algorithm has been used in this method. After detecting the eye, pixel position data from each eye is entered to multi-model Kalman filter. The purpose of this method is to predict the eye position in the next frame. After solving the equation of filter, estimation of eye position, pixel position in the next fame, proposed algorithm consider the accuracy of estimation based on eye data in frame. In proposed algorithm, the distance between the actual amount of eye position and the estimated amount is used for evaluation. If the estimated accuracy is appropriate, updated probability of each model and estimation of eye position is done next. Also in the case of higher uncertainty the proposed algorithm returns to eye tracking.

Understandably, multi-model Kalman filter is useful not only for improving the performance of eye tracking, but also for optimizing estimation errors of eye tracking and the existence of noise in measured data. The filter nature and Gaussian white noise in measured data are the reasons of this. The first step of eye tracking is detecting it. In this paper, two-level algorithm is used for detection of first position of eye. By two-level algorithm we mean the use of two steps for estimating eye position in a single frame of a film. It should be noted that said algorithm has almost a lot of volume of calculation and cannot be used in every frame. Each level is explained in the following. General schematic method of eye tracking is shown in Figure 2.

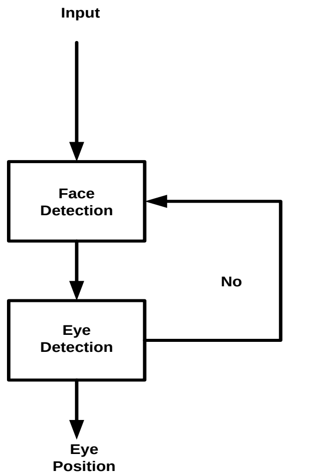


Figure 2. Used eye tracking algorithm

First level pursues identify the face in a frame. In other words, the pixel position of the face in an image is the output of first level. As mentioned before, the support vector machine is used for doing this work. For evaluating used algorithm in the first level, there is an image like a). The result of face detection (face level) is shown in b)**.** It is clear that the proposed algorithm can identify the face in the studied image well. In the first level, face position in the image is clear. The purpose of second level is to analyze the data of pixels which are in the face position for the face detection. Also support vector machine is used for this level. The result of second level is shown in c). In this image, the left eye has the horizontal pixel position 466 and vertical 265 and right eye has the horizontal pixel position 765 and vertical pixel 276. It should be noted that in those data, eye pixel position is used as multi-model Kalman filter entrance.

(a) (b) (c)

Figure 3. The process of personal algorithm of eye tracking: (a) Man image; (b) Recognition range of face; (c)The results of eye detection algorithm

2.2. Eye Tracking Algorithm

In this study, multi-model Kalman filter is used for estimating eye position in the next frame of film. Therefore, multi-model Kalman filter and its use ought to be explained. Multi-model Kalman filter estimator is a sub-optimal hybrid filter and it is very useful for hybrid estimation of target's position. The main strength of this method is the estimation of system's modes with several behavioral models and the ability to switch between them. In practice, we can take multi-model Kalman filter as a self-tuning filter with flexible bandwidth. Assume that the equation of state space of moving object is Equation 6.

$$\begin{aligned} x(k+1) &= F\left(k, m(k+1)\right)x(k) + G\left(k, m(k+1)\right)v\left(k, m(k+1)\right) \\ z(k) &= H(k)x(k) + w(k) \end{aligned} \tag{6}$$

In the Equation 6, $x \in \Re^{n_x}$ is object mode vector, $z \in \Re^{n_z}$ is measured output vector, $w \in \Re^{n_z}$ and $w \in \Re^{n_u}$ are independent white Gaussian noise vectors, andQ_v and R_w are covariance. Also $m(k)$ shows the mode of system, F represents system dynamic matrix and H represents the measurement matrix. Generally, various models have to be used to describe the system since there is no exact system model. In this method, we show the probability of mode I and K being used at the same as following: $M_i(k) = \{m(k) = m_i\}$. In the presented method, it is assumed that order of system's model changes follow a Markov Chain with the transition probability of Equation 7;

$$P\left\{m_j(k+1) \mid m_i(k)\right\} = P_{ij}(k) \tag{7}$$

Which, "$\sum_{j=1}^{r} P_{ij}(k) = 1,\ i = 1, 2, \dots\dots, r$" in above equation is correct based on Markov Chain Mathematics. Also in this method, a Markov transition matrix is used for describing the probability of the object in explained model. The probability of models is updated at the end of each algorithm repetition, and obtained results such are considered as weighting coefficient of each model. In short, a full cycle of implementation of the multi- model Kalman filter algorithm is as follows:

First Step: Mixing Probabilities

The probability of M_i model affecting the $k-1$ time is calculated based on the probability of M_j model affecting K time and is shown in this Equations 8 and 9.

$$P(M_i(k) \mid Z(k)) = \mu_i(k) \tag{8}$$

$$\begin{aligned} \mu_{i|j}\left(k-1 \mid k-1\right) &= P\left\{M_i(k-1) \mid M_j(k), Z^{k-1}\right\} \\ &= \frac{1}{\bar{c}_j} p_{ij}\,\mu_i(k-1) \end{aligned} \tag{9}$$

In Equation 9, the estimated probability for I-th mode is c^{-}_{j} which is calculated as in Equation 10.

$$\bar{c}_j = \sum_{i=1}^{r} p_{ij}\,\mu_i(k-1) \tag{10}$$

Second Step: Calculating Initial Mixing Values

According to previous estimations, for state variable and covariance matrix (in order $\widehat{x^{\iota}}\,(k-1|k-1)$ and $\widehat{x^{\iota}} = P^{i}(k-1|k-1)$ primal combination conditions are calculated as in Equation 11.

$$\begin{aligned} \hat{x}^{oj}\left(k-1 \mid k-1\right) &= \sum_{i=1}^{r} \hat{x}^{i}\left(k-1 \mid k-1\right) \mu_{i|j}\left(k-1 \mid k-1\right) \\ P^{oj}\left(k-1 \mid k-1\right) &= \sum_{i=1}^{r} \mu_{i|j}\left(k-1 \mid k-1\right) \left\{P^{i}\left(k-1 \mid k-1\right)\right\} + \\ &\left[\hat{x}^{i}(k-1 \mid k-1) - \hat{x}^{oj}(k-1 \mid k-1)\right]\left[\hat{x}^{i}(k-1 \mid k-1) - \hat{x}^{oj}(k-1 \mid k-1)\right]^{T} \end{aligned} \tag{11}$$

Third step: Calculation of Fitness Function for each Model First, by use of Kalman filter Equation 12

$$\begin{aligned} &\hat{x}_i\left(k \mid k-1\right) = F_i(k-1)\hat{x}_i(k-1 \mid k-1) + G(k-1)v(k-1) \\ &\hat{x}_i\left(k \mid k\right) = \hat{x}_i\left(k \mid k-1\right) + K_{f,i}(k)v_i(k) \\ &v_i(k) = z(k) - H_i(k)\hat{x}_i\left(k \mid k-1\right) \\ &P_i(k \mid k-1) = F_i(k-1)P_i(k-1 \mid k-1)F_i^{T}(k-1) + GQ(k)G^{T} \\ &S_i(k) = H_i(k)P_i(k \mid k-1)H_i^{T}(k) + R(k) \\ &K_i(k) = P_i(k \mid k-1)H_i^{T}(k)S_i^{-1}(k) \\ &P_i(k \mid k) = P_i(k \mid k-1) - K_i(k)S_i(k)K_i^{T}(k) \end{aligned} \tag{12}$$

Then, fitness function will be achieved for each model Equation 13;

$$\Lambda_j(k) = N\left[v_j(k); 0; S_j(k)\right] = \left|2\pi S_j(k)\right|^{-\frac{1}{2}} \times \exp\left[-0.5 v_j^{T}(k) S_j^{-1}(k) v_j(k)\right] \tag{13}$$

Forth Step: Model Probability

According to Equations 14 and 15, the probability of each model will be calculated.

$$c = \sum_{j=1}^{r} \Lambda_j(k)\bar{c}_j \tag{14}$$

$$\mu_j(k) = \frac{1}{c}\Lambda_j(k)\bar{c}_j \tag{15}$$

Fifth Step: Estimation of Modes and Calculating the Covariance Matrix

$$\hat{x}(k \mid k)=\sum_{i=1}^{r}\hat{x}^{i}(k \mid k)\mu_i(k) \tag{16}$$

$$P(k \mid k)=\sum_{i=1}^{r}\mu_i(k)\begin{Bmatrix} P^{i}(k \mid k)+\left[\hat{x}^{i}(k \mid k)-\hat{x}(k \mid k)\right] \\ \left[\hat{x}^{i}(k \mid k)-\hat{x}(k \mid k)\right]^{T} \end{Bmatrix} \tag{17}$$

The steps stated above, are for implementing the next step after measuring by sensor and by each sensor measurement, the steps are repeated. In Figure 4, general diagram block of multi-model Kalman filter method is shown.

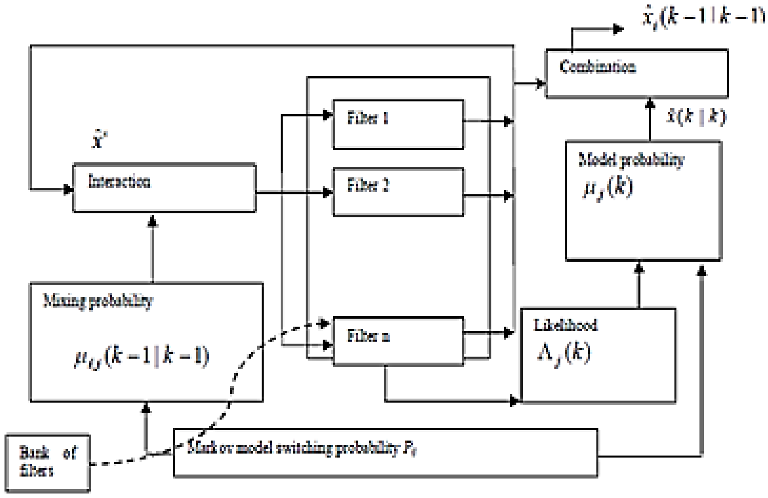


Figure 4. General block diagrams of multi model Kalman filter method

As was stated before, several models are used in this method. The models are divided into two main categories: Constant Speed Model and Constant Rotation Rate. In the constant speed model, there is fixed speed object and in constant rotation rate there is constant angular speed. Equation 18 elaborate the argument**.**

$$x_{t+1}=Fx_t+Gu_t$$

$$F_{CV}=diag\begin{bmatrix}1 & \Delta T\\ 0 & 1\end{bmatrix},\quad G=diag\begin{bmatrix}\frac{\Delta T^2}{\Delta^2 T}\\ \Delta T\end{bmatrix}\quad F_{CT}=\begin{bmatrix}1 & \frac{\sin\omega\Delta T}{\omega} & 0 & -\frac{1-\cos\omega\Delta T}{\omega}\\ 0 & \cos\omega\Delta T & 0 & -\sin\omega\Delta T\\ 0 & \frac{1-\cos\omega\Delta T}{\omega} & 1 & \frac{\sin\omega\Delta T}{\omega}\\ 0 & \sin\omega\Delta T & 0 & \cos\omega\Delta T\end{bmatrix} \tag{18}$$

Where $u_t : N\left(0, diag(\sigma_x^{\,2},\sigma_y^{\,2})\right)$.

In this equation, ΔT is the time of sampling and ω is the angular speed vector of object. Needless to say, all the models are linear.

3. Result and Discussion

The purpose of this study is to present eye tracking algorithm based on multi-model filter. Based on the previous arguments, there are several models instead of one model which are used in Kalman filter method. It is clear that the important weakness of Kalman filter is constant eye dynamic movement. In other words, the very reason of employing several models instead on one is optimizing the efficiency of proposed algorithm when it faces dynamic objects. In this section, we are to simulate multi-model Kalman filter while taking different models into account. It is crucial to say that in this simulation, two models of Constant Speed Model and Constant Rotation Rate are considered.

$$\begin{cases}\bar{x}_{k+1}=A_i\bar{x}_k+B_i v_k\\ y_k=C_i\bar{x}_k+\omega_k\end{cases}$$

$$\bar{x}_k=[x_k\dot{x}_k y_k\dot{y}_k]^T \tag{19}$$

$$\mathcal{A}_1=\begin{bmatrix}\mathrm{A}_1 & 0\\ 0 & \mathrm{A}_1\end{bmatrix}, \mathrm{A}_1=\begin{bmatrix}1 & \mathrm{T}\\ 0 & 1\end{bmatrix}, \mathcal{B}_1=\begin{bmatrix}\mathrm{B}_1 & 0\\ 0 & \mathrm{B}_1\end{bmatrix}\qquad \textit{Object dynamic}$$

$$B_1 = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \qquad \textit{Constant speed matrix model} \tag{20}$$

$$\bar{x}_k = [x_k \dot{x}_k \ddot{x}_k y_k \dot{y}_k \ddot{y}_k]^T$$

$$\mathcal{A}_2 = \begin{bmatrix} A_2 & 0 \\ 0 & A_2 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}, \mathcal{B}_2 = \begin{bmatrix} B_2 & 0 \\ 0 & B_2 \end{bmatrix}, B_2 = \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} \qquad \textit{Constant acceleration} \tag{21}$$

The results of the algorithm simulation are explained in the following and are shown in Figure 5. This model follows the object very well.

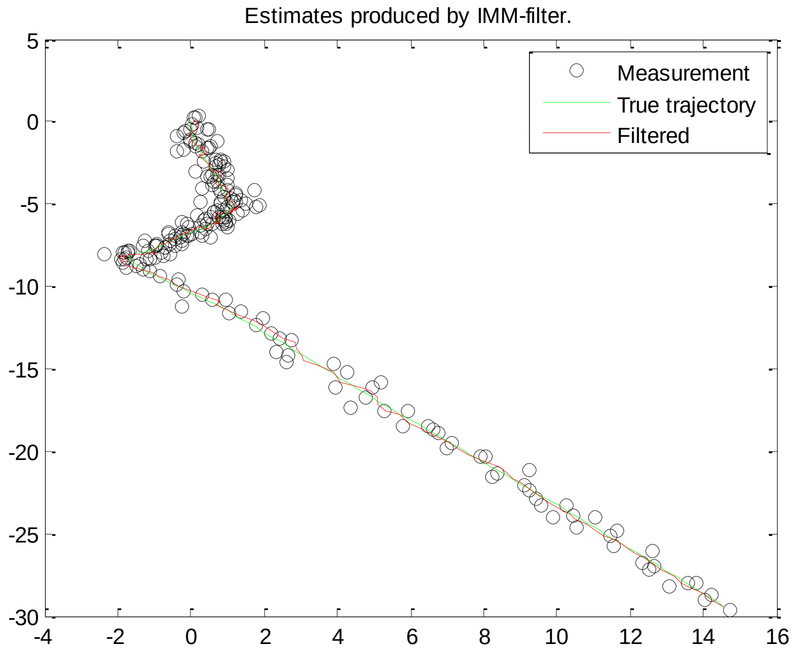


Figure 5. Results of multi-model interference filter simulation

Simulation

The purpose of this section is to consider the proficiency of proposed algorithm for eye tracking in an image in different frames. As explained before, there are different models for eye tracking such as constant speed, constant acceleration, constant rotation, and constant mass. Their efficiencies are different based on the type of object changes. Since we set to track eye, constant speed and constant acceleration models are used in this paper. Considering previous data and also covariance matrix of noise measurement, for constant speed Equation 22 and for constant acceleration Equation 23;

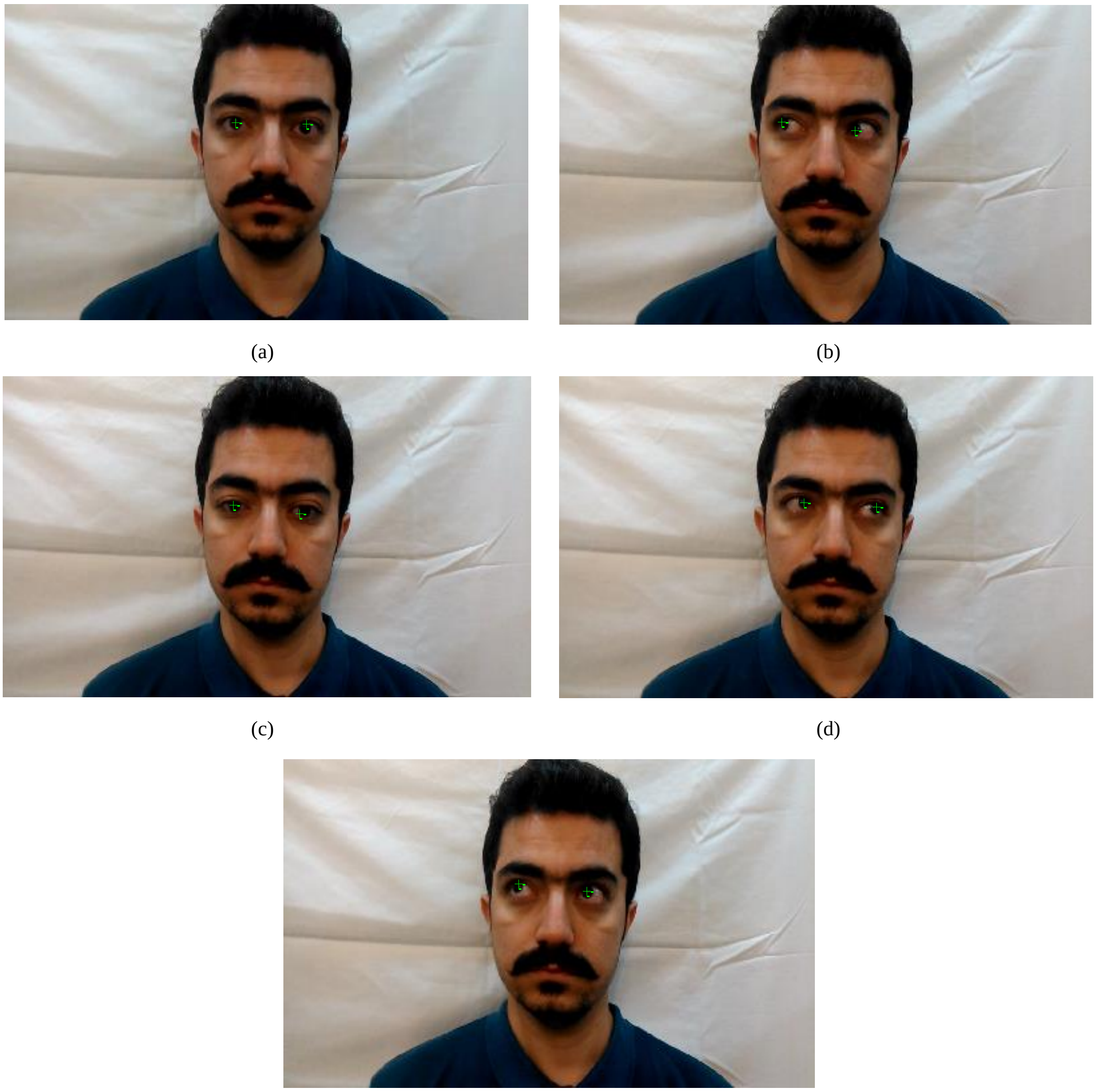
$$R = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}; \qquad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}$$

$$R = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}; \qquad Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix} \tag{23}$$

In the following part, we consider the efficiency of proposed algorithm in different frames of a film for eye tracking.

First Scenario:

The purpose is to trace the pupil by constant face. The face is fixed and the pupils have an anti-clockwise circular motion. Different frames and the results are shown in Figure 6. It is clear that the proposed algorithm could trace the pupil well.



(a) (b)

(c) (d)

(e)

Figure 6. The results of eye tracking in the first scenario: a) frame 1; b) frame 50; c) frame 100; d) frame 150; c) frame 200

In Figures 7 and 8, the errors of estimating eye position to the real mode by proposed algorithm and Kalman filter are shown. It should be noted that in this figure, measurement of the horizontal axis is pixel.

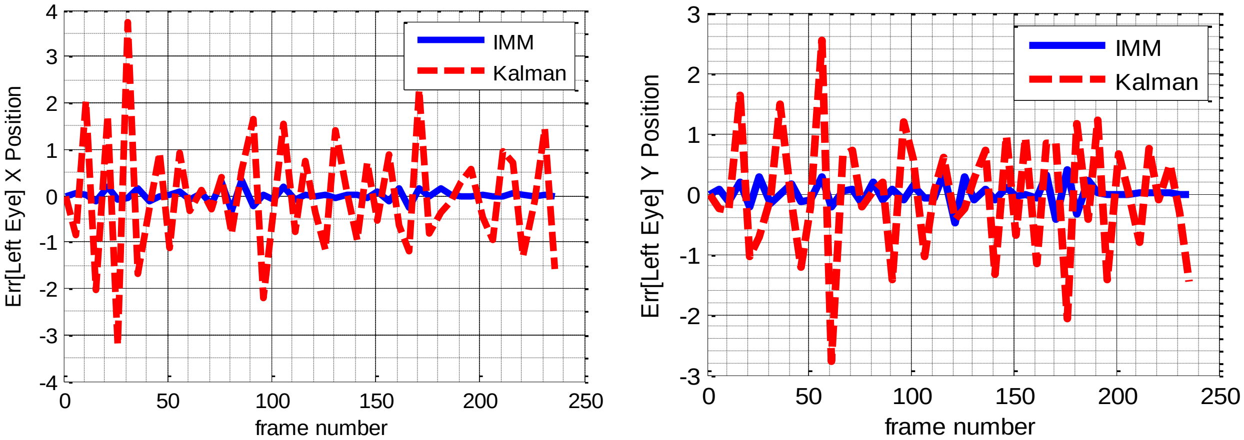


Figure 7. The error of left eye estimation of position

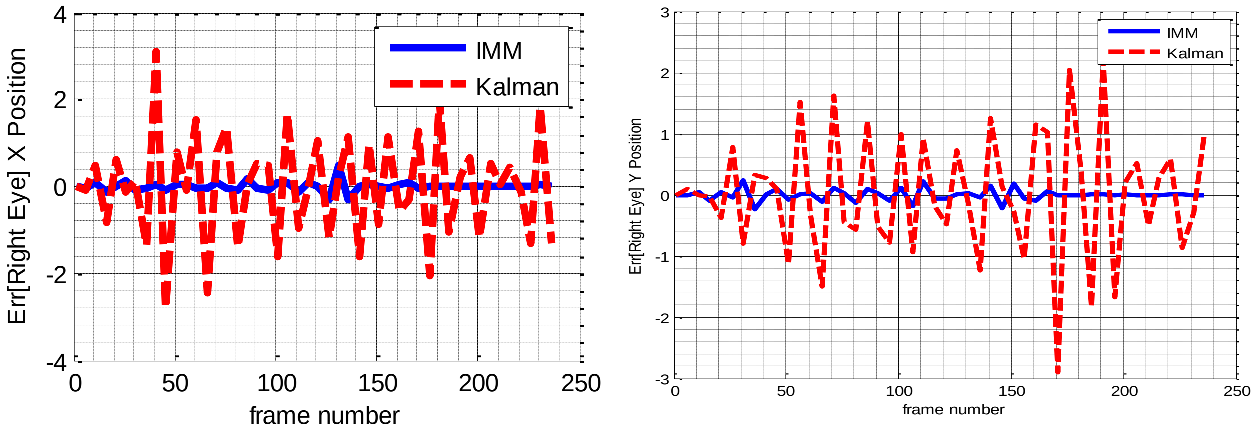


Figure 8. The error of right eye estimation of position

Finally, in Table 1 compares results of noise measurement on eye position to Kalman filter.

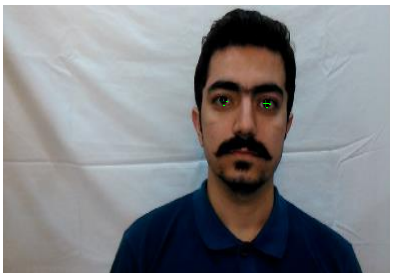
Table 1. Consideration of the efficiency of Kalman filter in first scenario

Measurement Noise	Proposed Filter		Kalman Filter	
	Left	Right	Left	Right
%80 Nominal value	6.62	8.743	65.26	62.184
%120 Nominal value	8.02	12.53	67.57	93.66
%200 Nominal value	24.18	16.537	95.32	111.88
%300 Nominal value	39.54	19.26	149.10	148.48

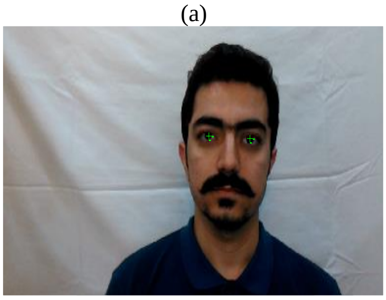
Second Scenario:

The purpose is eye tracking by animated face. At first, face moves to right; and then left. The results of this simulation are shown in different frames. It is clear that the proposed model could trace the pupil very accurately.

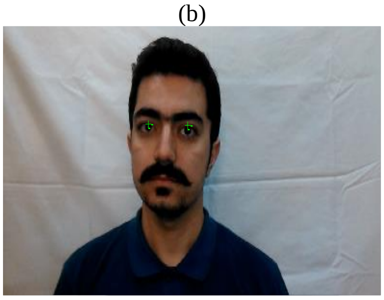
(a)



(b)



(c)



(d)

(e)

Figure 9. The results of eye tracking in the second scenario: a) frame 1; b) frame 50; c) frame 100; d) frame 150; c) frame 200

Figures 10 and 11 show how off the trace has been compared to real state of the eye.

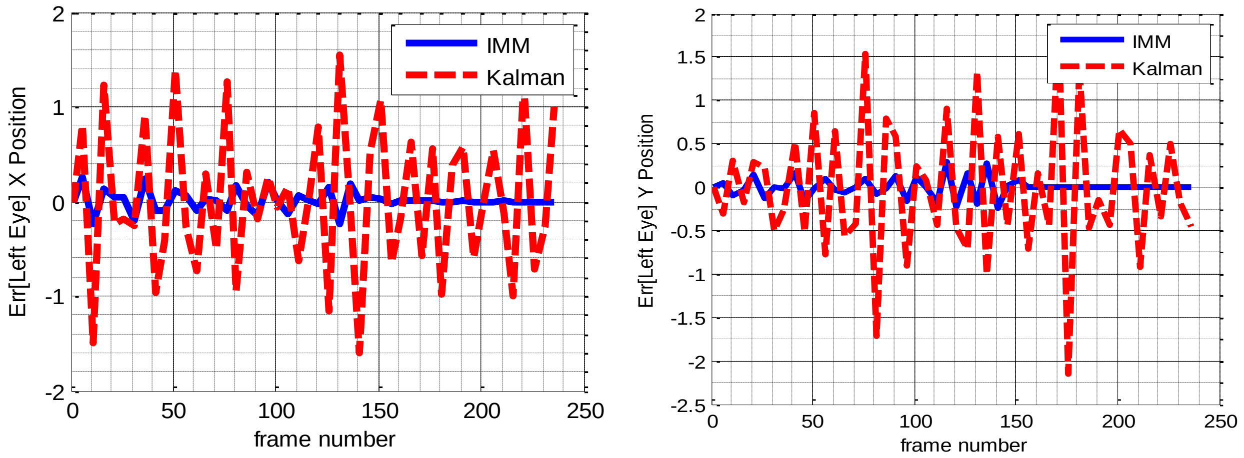


Figure 10. The error of left eye estimation of position

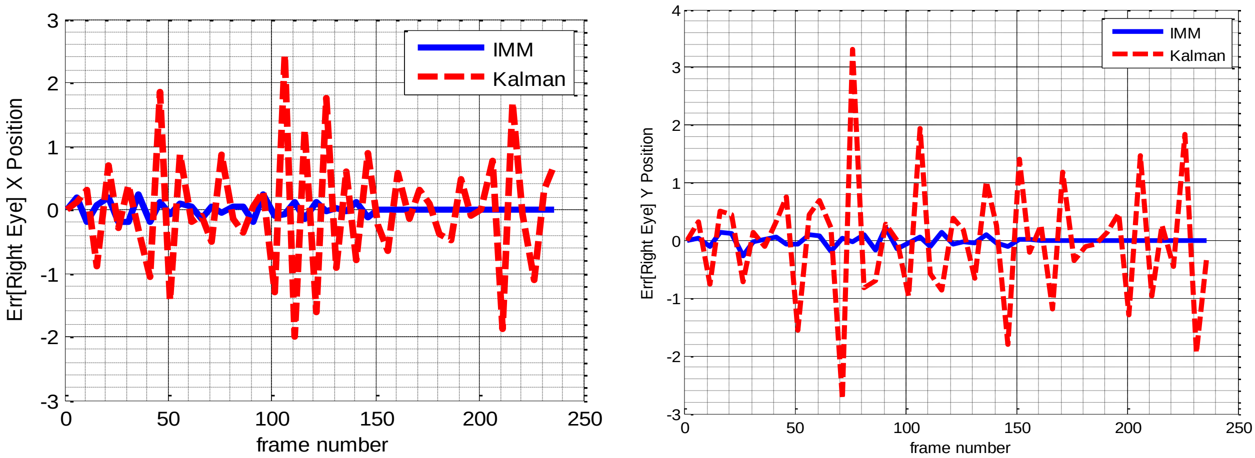


Figure 11. The error of right eye estimation of position

Finally, the results of the different noise measurement on eye position are shown in Table 2. Clearly enough, proposed method works a lot better than Kalman filter.

Table 2. Consideration of proposed filter in the second scenario

Measurement Noise	Proposed Filter		Kalman Filter	
	Left	Right	Left	Right
%80 Nominal value	8.5	8.24	59.7	79.5
%120 Nominal value	8.43	8.85	63.3	83.35
%200 Nominal value	22.93	13.55	95.16	102.59
%300 Nominal value	49.45	26.22	135.69	150.45

Third Scenario:

The purpose is eye tracking by animated camera. The position of face is fixed but the camera is moving. The results are shown in and this model has high efficiency. Results are shown in Figure 12.

(a)

(b)

(c)

(d)

(e)

Figure 12. The results of eye tracking in the first scenario: a) frame 1; b) frame 50; c) frame 100; d) frame 150; c) frame 200

Figures 13 and 14 show how off the trace has been compared to real state of the eye.

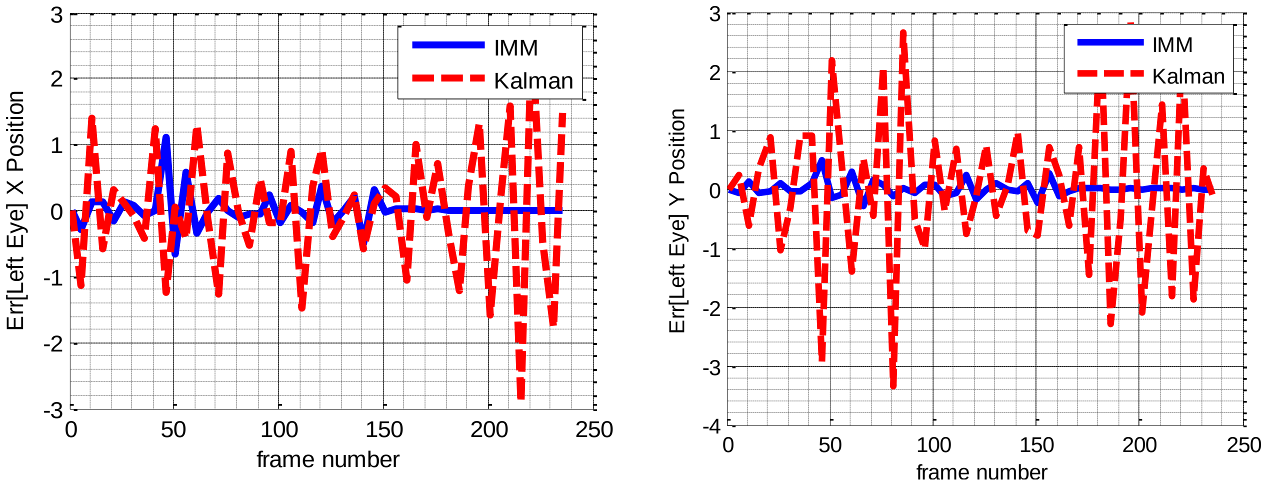


Figure 13. The error of left eye estimation of position

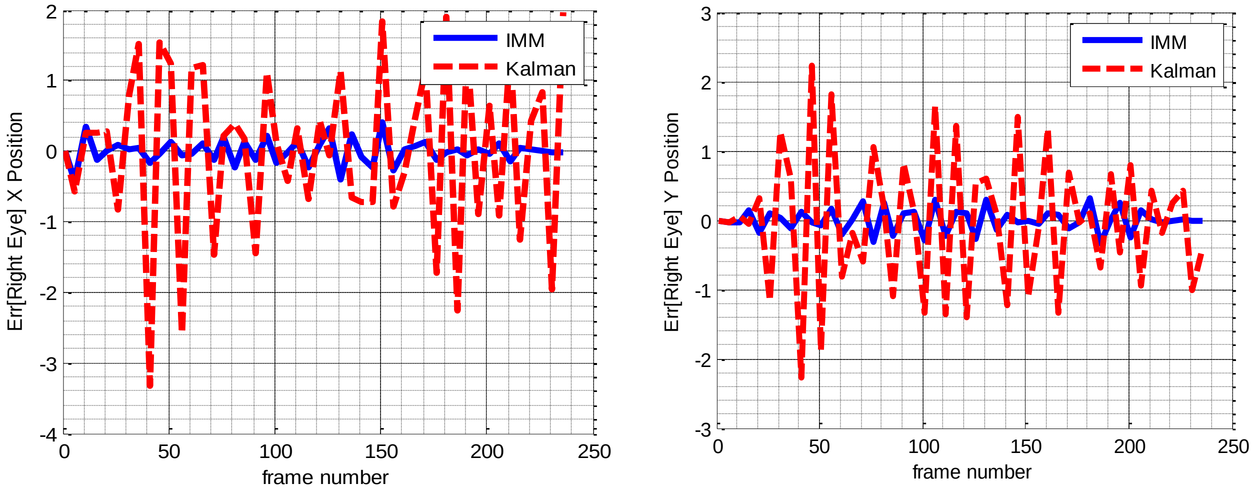


Figure 14. The error of right eye estimation of position

At the end, the results of the different noise measurements on eye position are provided in Table 3. The efficiency of the proposed method is considerably high.

Table 3. Consideration of proposed filter in the third scenario

Measurement Noise	Proposed Filter		Kalman Filter	
	Left	Right	Left	Right
%80 Nominal value	9.2325	8.81	64.96	44.24
%120 Nominal value	10.72	9.65	73.86	72.58
%200 Nominal value	17.53	11.21	86.29	79.35
%300 Nominal value	25.97	20.31	94.32	104.52

4. Conclusion

In this study, the results of the proposed algorithm simulation for eye tracking in different frames were studied. In the Kalman filter, there is just one model, but in the multi-model Kalman filter, there are different models for estimating the object, of which two models, constant speed and constant acceleration, are used in this paper. Furthermore, a support vector machine is also used in this study for identifying the eye. It is clear that the proposed algorithm could trace the eye's position in the film with high accuracy. There are some important notes for readers to discuss the proposed models: Multi-model Kalman has better efficiency than the Kalman filter; the proposed model does not need an accurate model of object dynamics (eye); it can be used in real time.

5. Declarations

5.1. Author Contributions

S.H.Z.B. and S.T. contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Informed Consent Statement

Informed consent was obtained from all individual participants included in the study.

5.5. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6. References

[1] Xiong, J., Xu, W., Liao, W., Wang, Q., Liu, J., & Liang, Q. (2013). Eye control system based on ameliorated though transform algorithm. IEEE Sensors Journal, 13(9), 3421–3429. doi:10.1109/JSEN.2013.2262934.

[2] Yuille, A. L., Hallinan, P. W., & Cohen, D. S. (1992). Feature extraction from faces using deformable templates. International Journal of Computer Vision, 8(2), 99–111. doi:10.1007/BF00127169.

[3] Xie, X., Sudhakar, R., & Zhuang, H. (1994). On improving eye feature extraction using deformable templates. Pattern Recognition, 27(6), 791–799. doi:10.1016/0031-3203(94)90164-3.

[4] Shah, J., Sharif, M., Raza, M., & Azeem, A. (2013). A survey: Linear and nonlinear PCA based face recognition techniques. International Arab Journal of Information Technology, 10(6), 536–545.

[5] Huang, W., & Mariani, R. (2000). Face detection and precise eyes location. Proceedings - International Conference on Pattern Recognition, 15(4), 722–727. doi:10.1109/icpr.2000.903019.

[6] Huang, J., & Wechsler, H. (1999). Eye detection using optimal wavelet packets and radial basis functions (RBFs). International Journal of Pattern Recognition and Artificial Intelligence, 13(7), 1009–1025. doi:10.1142/S0218001499000562.

[7] Reinders, M. J. T., Koch, R. W. C., & Gerbrands, J. J. (1996). Locating facial features in image sequences using neural networks. Proceedings of the International Conference on Automatic Face and Gesture Recognition, 230–235. doi:10.1109/afgr.1996.557269.

[8] Wang, C., Shi, F., Xia, S., & Chai, J. (2016). Real-time 3D eye gaze animation using a single RGB camera. ACM Transactions on Graphics, 35(4), 1–14. doi:10.1145/2897824.2925947.

[9] Vrânceanu, R., Florea, C., Florea, L., & Vertan, C. (2015). Gaze direction estimation by component separation for recognition of Eye Accessing Cues. Machine Vision and Applications, 26(2–3), 267–278. doi:10.1007/s00138-014-0656-8.

[10] Dong, C., Wang, X., Pei-Hua, C., & Pu-Liang, Y. (2015). Eye Detection Based on Integral Projection and Hough Round Transform. Proceedings - 2015 IEEE 5th International Conference on Big Data and Cloud Computing, BDCloud 2015, 252–255. doi:10.1109/BDCloud.2015.34.

[11] Morcego, B., Argilés, M., Cabrerizo, M., Cardona, G., Pérez, R., Pérez-Cabré, E., & Gispets, J. (2016). Blinking supervision in a working environment. Journal of Biomedical Optics, 21(2), 025005. doi:10.1117/1.jbo.21.2.025005.

[12] Lescroart, M. D., Kanwisher, N., & Golomb, J. D. (2016). No evidence for automatic remapping of stimulus features or location found with fMRI. Frontiers in Systems Neuroscience, 10(Jun), 53. doi:10.3389/fnsys.2016.00053.

[13] Amiel, F., Barry, B., Trocan, M., & Swynghedauw, M. (2015). Real time image compression for eye tracking applications. 2015 IEEE 6th Latin American Symposium on Circuits and Systems, LASCAS 2015 - Conference Proceedings, 1–4. doi:10.1109/LASCAS.2015.7250463.

[14] Zhu, Z., & Ji, Q. (2007). Novel eye gaze tracking techniques under natural head movement. IEEE Transactions on Biomedical Engineering, 54(12), 2246–2260. doi:10.1109/TBME.2007.895750.

[15] Yoo, D. H., & Chung, M. J. (2005). A novel non-intrusive eye gaze estimation using cross-ratio under large head motion. Computer Vision and Image Understanding, 98(1), 25–51. doi:10.1016/j.cviu.2004.07.011.

[16] Zhu, Z., Ji, Q., Fujimura, K., & Lee, K. (2002). Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination. In Proceedings - International Conference on Pattern Recognition, IEEE, 16(4), 318–321. doi:10.1109/icpr.2002.1047460.

[17] Zhang, Z., & Zhang, J. (2010). A new real-time eye tracking based on nonlinear unscented Kalman filter for monitoring driver fatigue. Journal of Control Theory and Applications, 8(2), 181–188. doi:10.1007/s11768-010-8043-0.

[18] Li, Y., Deng, J., Wu, Q., & Wang, Y. (2021). Eye-Tracking Signals Based Affective Classification Employing Deep Gradient Convolutional Neural Networks. International Journal of Interactive Multimedia and Artificial Intelligence, 7(2), 34. doi:10.9781/ijimai.2021.06.002.

[19] Barbara, N., Camilleri, T. A., & Camilleri, K. P. (2020). EOG-Based Ocular and Gaze Angle Estimation Using an Extended Kalman Filter. ACM Symposium on Eye Tracking Research and Applications. doi:10.1145/3379156.3391357.

[20] Wu, Q., Dey, N., Shi, F., Crespo, R. G., & Sherratt, R. S. (2021). Emotion classification on eye-tracking and electroencephalograph fused signals employing deep gradient neural networks. Applied Soft Computing, 110, 107752. doi:10.1016/j.asoc.2021.107752.