

Original Article

A Fast, Robust, Automatic Blink Detector

Javad Sayahzadeh^{1*}, Hamidreza Pourreza¹, Javad Salehi Fadardi²

Abstract

Introduction

“Blink” is defined as closing and opening of the eyes in a small duration of time. In this study, we aimed to introduce a fast, robust, vision-based approach for blink detection.

Materials and Methods

This approach consists of two steps. In the first step, the subject's face is localized every second and with the first blink, the system detects the eye's location and creates an open-eye template image. In the second step, the eye is tracked, using sum of squared differences (SSD). This system can classify the state of the eyes as open, closed, or lost, using the SSD-based classifier. If the eyes are closed as in usual blinking, the blink will be detected. To classify eyes as closed or open, two adaptive thresholds were proposed; therefore, factors such as the subject's distance from the camera or environment illumination did not affect the system performance. In addition, in order to improve system performance, a new feature, called "peak-to-neighbors ratio", was proposed.

Results

The accuracy of this system was 96.03%, based on the evaluation on Zhejiang University (ZJU) dataset, and 98.59% in our own dataset.

Conclusion

The present system was faster than other systems, which use normalized correlation coefficient (NCC) for eye tracking, since time complexity of SSD is lower than that of NCC. The achieved processing rate for ZJU dataset was 35 fps.

Keywords: Eye Localization; Blink Detection; Sum of Squared Differences

1- Machine Vision Lab., Computer Eng. Department, Ferdowsi University of Mashhad, Iran

*Corresponding author: Tel: +985133030383 E-mail: j.sayahzadeh@gmail.com

2- Faculty of Educational and Psychological Sciences, Ferdowsi University of Mashhad, Iran

1. Introduction

In recent years, many studies have evaluated image processing and machine vision fields to identify human biometrics. Lip detection, eye detection, mouth detection, face detection, face tracking, facial expression recognition, face recognition, finger print recognition, iris recognition, and blink detection are examples of these efforts, most of which are face-related.

“Blink” is defined as closing and opening of the eyes in a small duration of time, and is necessary for the eyes. Each blinking spreads the tear film over the eyes and clears the corneas. Blinking rate can be affected by different factors such as eye fatigue, injury, medication use, disease, and stimulation (light reflection or an object’s impact on the eyes) [1].

Blink detection allows measurements of eyes’ blinking rate and closure duration. Blink detection approaches are divided into electro ocular-based and image processing-based approaches. The electro ocular approaches are intrusive since they require the user to place recording electrodes near the eyes [2]. On the other hand, image processing approaches are non-intrusive, though most of them cannot detect blinks in real time. Therefore, if we can increase the speed of image processing approaches, it will be more convenient for users to avoid intervening objects, applied in former approaches.

Blink detection has diverse applications, e.g., helping disabled individuals control the clicking of a mouse, driver drowsiness detection for reducing accidents, and fatigue detection for preventing monitor-related eye syndrome.

One of the major problems of blink detection methods is datasets, used for testing. In most datasets, subjects’ eyes have voluntary behaviors, which differ from spontaneous movements. Therefore, systems, which use this type of datasets, are only applied in laboratories and are not applicable in the real world.

Herein, we review recent studies in relation to blink detection and its application.

Grauman et al. proposed a video-based human-computer interface, named “Blink Link” [3, 4]. Application of this tool helps individuals with disabilities to control mouse clicks. This system

detects blinks and eye closure duration. Long blinks act as clicks and short blinks are ignored. First, frame subtraction is performed to detect motion regions. Then, a pre-trained open-eye template is selected from the motion regions, using Mahalanobis distance. Eyes are tracked and correlation scores between the actual eye and the corresponding open-eye template are used to detect blinks. The accuracy of this system is reported as 95.6%, based on the used dataset. However, this tool requires pre-training for different distances from the camera for more robustness. A disadvantage of this system is that changing the camera position requires the system to be retrained. Furthermore, the system will not be effective if it is used on different eye forms and sizes.

Królak et al. presented a vision-based system for the detection of voluntary eye blinks and evaluated its implementation as a human-computer interface for people with disabilities [5]. In their proposed system, face detection is implemented by means of Haar-like features and a cascade of boosted tree classifiers. The position of the eyes in the face image is detected, based on certain geometrical dependencies, known for human face. The image of the extracted eye region is further pre-processed for performing eye-blink detection. Correlation coefficient changes in time are analyzed in order to detect voluntary or spontaneous eye blinks. If the coefficient value is lower than the predefined threshold value (TL) for any two consecutive frames, the onset of the eye blink is detected. The accuracy of this system in good illumination conditions was reported as 95.35%, despite its lower accuracy in poor illumination conditions (75.02%).

Morris et al. proposed a computer operation method, controlled by the subject’s head and eyes [6]. This system tracks the user’s motions and translates head motions to mouse cursor movements and facial expression changes to keys pressings. This system uses spatial filtering for face localization in image sequences. Then, it tracks the face and finds the eyes, based on blinking, using variance shadow maps. It then detects contours surrounding the eyes and tracks

them using the modified Lucas-Kanade tracker [7].

In a study by Heishman *et al.*, detection of blinks with ambiguous movements (e.g., partly closed) was introduced [8]. The introduced system detects blinks using image flow, and the image flow analysis defines the magnitude and direction of blink motions. This system uses a deterministic, finite state machine, and calculates blink parameters (e.g., blink rate) by using the motion data. This system is a modification of the blink detection system, introduced by Bhaskar *et al.*, which subtracts successive frames for blink detection [9].

In addition, Pan *et al.* proposed a method that uses blinking type as a discriminative parameter of live face and photograph in face recognition [10]. Based on the boosting algorithm, Pan proposed a discriminative feature for eye imaging, named "eye closity" which measures the degree of eye closure. The accuracy of this system for the detection of both eyes was 95.7 %, based on its application on Zhejiang University (ZJU) dataset. Divjak *et al.* introduced a model for the prevention of computer vision syndrome (CVS) [11]. For a more detailed description of CVS, you can refer to the references [12]. This model uses a machine vision system for detecting the subject's blinking and finally detects the eyes' fatigue. This method captures subjects by using a 320×240 webcam. It detects abnormal behaviors of the eyes, based on blinking patterns. This system has 96±7 percentage of accuracy.

Toricelli *et al.* introduced a blink-based method for eye localization [13]. The average accuracy reported for this system was 95.7%, using ZJU dataset. Similarly, the goal of the blink detection system by Bhaskar *et al.* was eye localization [9]. Danisman also presented a system to monitor drivers' drowsiness and prevent accidents by monitoring blink duration variations [14]. This system can detect the eyes' state by using only one image frame. This system localizes the face and the eyes, using Viola-Jones face detector and neural network-based eye detector, respectively. After adjustment the face orientation by eye direction, the eye region is detected and then,

contrast stretching is performed on a rectangular region of the eye.

This system subtracts the upper and lower halves of the eye region and calculates the cumulative sum of differentiated results. If the sum is higher than the specified threshold, it means that the eyes are open; otherwise, they are considered closed. This system does not work well in the presence of glasses or illumination variations. Danisman's blink detector was tested on ZJU dataset and 94% accuracy with 1% false positive rate (FPR) was reported [14].

Also, Kim *et al.* introduced a system which is a visual fatigue monitoring system, based on eye movements and eye blink detection [15]. The proposed system is equipped with an infrared single camera and an infrared light source. The pupil can be detected by applying a binary threshold to the Purkinje image. The threshold is automatically selected by two constraints, which are the eccentricity of ellipse fitting and the size of the pupil. Finally, the total of eye movements and the number of eye blinks are measured, using pupil positions.

Recently, Xu *et al.* presented a blink detector system [16]. In their system, first, face detection and face motion tracking are performed and then, a three-dimensional cross model is created. For this purpose, the middle of the two eyebrows and other four points on the face are selected as the endpoints. To track the eyes, the geometrical data are used. Finally, according to the gray-level histogram of eye image, the number of pixels, which are darker than a certain threshold, is obtained, and open/closed eye detection is performed. Xu *et al.* used a participant with dark skin. This algorithm obtained an overall accuracy of 99.24% after being tested on a dataset, which included videos of 20 black men.

Recently, Khan *et al.* presented a system to determine the appropriate length for a text line, based on blinking [17]. They mentioned that while reading a computer screen, the reader loses his/her focus first at the end of each line and second, while blinking in the middle of the line. This method can determine the optimal length of the line based on blinking at the end of the line. By doing so, they reduced the points of losing

focus to the end of each line at which the person blinks. However, the proposed system was only applied for women.

The proposed eye-blink detection approach is described in section 2. Study results are discussed in section 3, and conclusion is given in section 4.

2. Materials and Methods

The overall structure of our proposed blink detector is shown in Figure 1. This detector consists of two steps. In the first step, the subject's face is detected and the face motions are analyzed. After subtracting successive frames, binary image generation, morphological filtering (opening and closing), connected pair labeling, and invalid-pair elimination are performed; if a blink occurs, an open-eye template image will be generated.

In the second step, first, the search region is formed, based on the eye location. Then, dissimilarity using sum of squared differences (SSD) criterion between the open-eye template and search region (in which the eyes are closed) is calculated. According to the dissimilarity value, open- and closed-eye threshold values are calculated. Then, successive frames are fetched from the video and dissimilarity between the open-eye template and the search region is calculated by SSD criterion; afterwards, the state of the eye (closed, open, or lost) is recognized. In the following section, we describe all stages of blink detection (Figure 2).

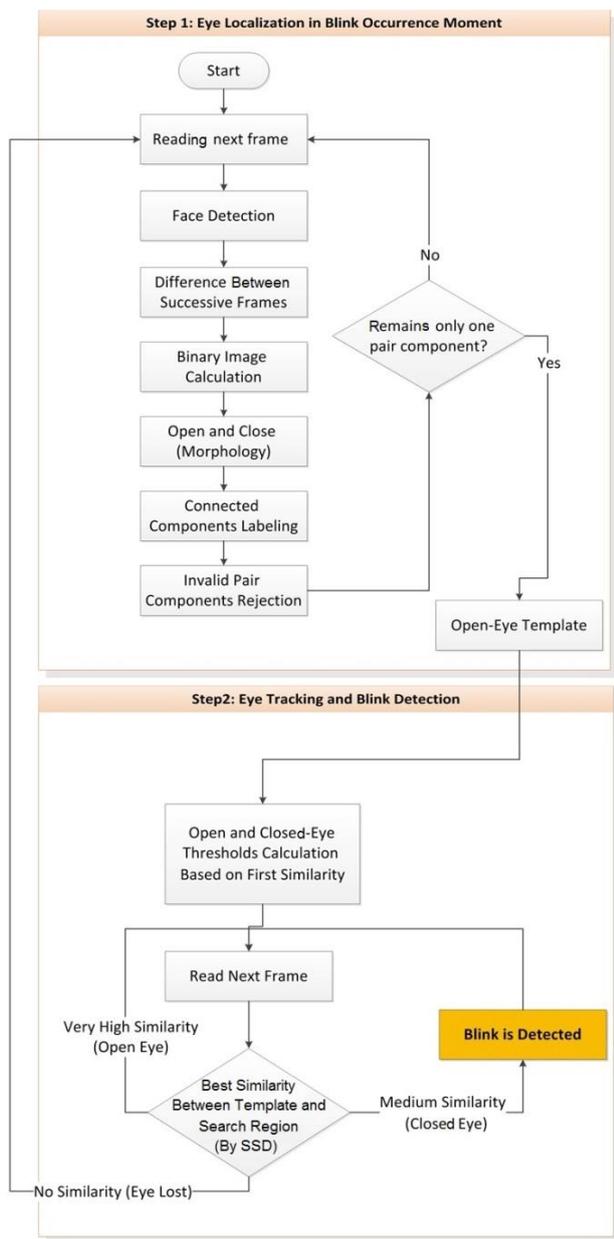


Figure 1. An overview of the executive stages of blink detector, comprised of two steps

2.1. Step1: Eye localization

The goal of this step is to find the position of the eyes on the face and to generate an open-eye template image as an input for the second step. In the following section, we describe all the stages of this step.

2.1.1. Face detection

At the beginning of the first step, we used a robust, real-time, and well-known face detector, named Viola-Jones object detector [18]. This face detector has been already applied in many previous studies [5, 14, 19]. The reason for using this face detector is to reduce the number of computations in the eye localization stage.

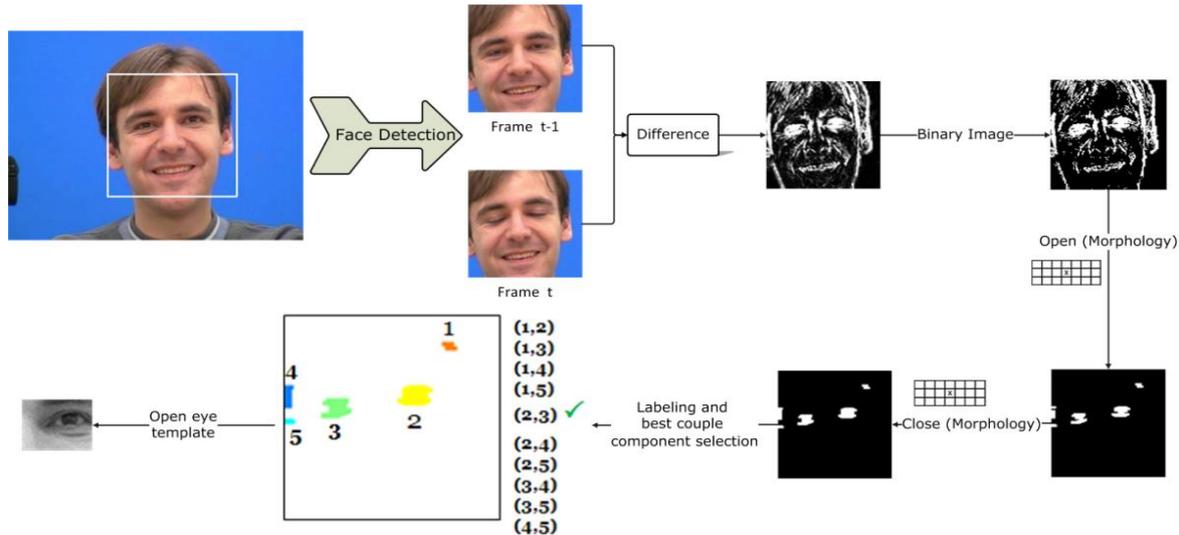


Figure 2. Stages of blink detection including the first step along with the relevant images



Figure 3. A face detected by Viola-Jones face detector

In Figure 3, an image of our dataset, called spontaneous blink detector (SBD), and face region, which is found by Viola-Jones face detector, is shown. As it can be seen, the number of pixels within the face region is fewer than that of the whole image.

Forward and backward head movements do not eliminate the face from the rectangle within one second; however, sided head movements cause the face to fall out of the rectangle surrounding the face. Since sided head movements rarely occur, we chose a one-second interval for face localization.

2.1.2. Subtraction of Successive Frame Images

By subtracting the current frame image from previous frame images, the moving regions can be defined. Due to changes in the eye at blinking instances, after subtracting the face image (in which the eyes are open) from the image of the next frame (where the eyes are closed), an image

is obtained in which pixels within moving regions are near to white (value 1) and those in non-moving regions are near to black (value 0). Then, by applying a square operator on the subtracted image, differences become apparent. These two operations are performed by applying the following equation:

$$I_d(x, y, t) = (I(x, y, t) - I(x, y, t - 1))^2 \quad (1)$$

where $I_d(x, y, t)$ is the pixel intensity of squared image difference at point (x, y) in frame t and $I(x, y, t)$ is the pixel intensity of the current frame image at point (x, y) in frame t .

An example of the subject's face images in two successive frames is shown in Figure 2. White pixels indicate movements occurring on those pixels and black pixels show no movements.

2.1.3. Binary Image Generation

After subtracting successive frames, binary image generation is carried out. For this purpose, based on the subtracted image, a threshold for binary image generation is obtained using the Otsu algorithm. Then, the binary image is generated by:

$$I_b(x, y, t) = \begin{cases} 0 & , I_d(x, y, t) < Th_d(t) \\ 1 & , \text{Otherwise} \end{cases} \quad (2)$$

where I_b is the binary image, x and y are pixel coordinates, and Th_d is the threshold that is obtained using Otsu algorithm. A binary image is shown in Figure 2.

2.1.4. Image Enhancement Using Morphological Operators

For reducing noise in the binary image, which may be produced due to various reasons such as source light variations, image edge intensity, and subject’s body movements, we applied two morphological operators (eye opening and closing, consecutively).

The structure element (SE), which has been used for these two operators, is a rectangle of 3×7 pixels (Figure 4). SE shape is selected a little smaller than the eye’s size in the binary image. However, the size of the SE is enlarged, based on the source image size.

By applying an opening operator on the binary image, an image is produced with fewer details. In the opening operation, the pieces of binary image, which do not fit in SE, are removed; thus, only those pieces which are larger than SE remain in the resulting image (Figure 2). In other words, binary image noises are removed by applying the opening operator. An example of applying this operator is presented in Figure 2.

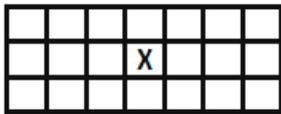


Figure 4 . Morphological structure element for opening and closing operators

For the compensation of size reduction of pieces in the opening operation, we applied a closing operator. By closing the image, pieces, which are close together, coalesce; this process also fills cavities and smoothes the surroundings. For an example of a closing image, see Figure 2.

2.1.5. Piece-pair labeling and selection

After applying morphological operators, some pieces may remain in the resulting image (see Figure 2), and among these pieces, two may belong to the eyes’ location. All the remaining pieces are then labeled. The labels are numbered from one up to the total number of pieces (see Figure 2). These labeled pieces are checked in a piece-pair manner to determine if they belong to real eyes or not. The number of states that must be checked is calculated by:

$$C(n, 2) = \frac{n!}{2!(n-2)!} \tag{3}$$

where n is the number of pieces. For instance, in Figure 2 in which five pieces remain after the morphological image enhancement, 10 piece pairs are checked.

For the selection of the best pair, which belongs to real eyes, all pairs are checked to see if they meet the following criteria:

1- The width of the first piece must be less than three times of the second piece width and the width of the second piece must be less than three times of the first piece width :

$$1/3 \leq \frac{cw_1}{cw_2} \leq 3 \tag{4}$$

Where cw_1 and cw_2 are the widths of the first and second pieces, respectively.

2- The height of the first piece must be less than three times of the second piece height and also the height of the second piece must be less than three times of the first piece height:

$$1/3 \leq \frac{ch_1}{ch_2} \leq 3 \tag{5}$$

Where ch_1 and ch_2 are the heights of the first and second pieces, respectively.

3- The width of each piece must be greater than its height and less than four times of its height:

$$1 \leq \frac{cw_i}{ch_i} \leq 4 \tag{6}$$

where cw_i is the width of the i piece and ch_i is the height of the i piece.

4- The horizontal difference between the centers of two pieces must be at most two times of the vertical difference:

$$\frac{\lambda_w}{\lambda_h} \leq 2 \tag{7}$$

where λ_w and λ_h are the horizontal and vertical distances between the centers of two pieces, respectively.

5- The vertical distance between the centers of two pieces must be greater than the sum of the width of two pieces and less than three times of the sum of two pieces’ widths.

$$1 \leq \frac{\lambda_w}{cw_1 + cw_2} \leq 3 \tag{8}$$

All of the abovementioned criteria are based on the geometrical features of the eye. For a better

understanding, the terms, used in the above equations, are defined in Figure 5.

If only one piece-pair could satisfy all of the above criteria, it relates to the eyes.

The value of these criteria is obtained by applying each criterion on various and numerous valid/invalid samples, and finally, the best one with the lowest error is found. Therefore, the criteria are designed robustly and accurately.

One of the advantages of the mentioned criteria is their application on various video sizes, since their values are relative to video size and are not constant.

Both eyes show the same behavior at a blinking instance; therefore, for reducing the number of calculations, we continue step 2 with only one eye. One of the two pieces, which has a larger area, is passed to the second step.

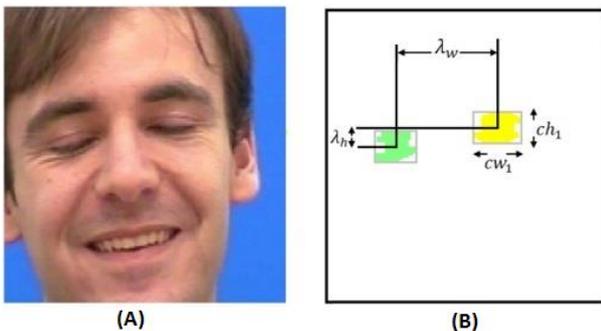


Figure 5. Parameters of the best piece-pair selection; (A) subject's face (his eyes are closed during blinking); (B) one piece-pair along with the definition of parameters

2.1.6. Open-eye template image

After piece-pair selection, the eye template image is created. In the frame of blinking (t), eyes are closed; hence, in five frames earlier (frame $t - 5$), the eyes are open and are perfect for open-eye template generation in a 30 fps video. If this number is small, the eyes are likely to be closed and if this number is large, the subject's movement leads to selecting a bad location and the eyes may be closed due to the previous blink. The open-eye template image size is chosen a little larger than the selected piece (best piece-pair selection). A generated open-eye template image, which is the output of the first step of our proposed blink detection, is shown in Figure 2. In

addition, some open-eye template images, which are obtained from various datasets, are shown in Figure 6.



Figure 6. Open-eye template examples with various sizes of the eye, luminance changes, with or without glasses

2.2. Step 2: Eye tracking and blink detection

The input of this step is the output of the first step, namely the open-eye template image. In this step, eye tracking and blink detection are performed simultaneously. Based on the dissimilarity value between the open-eye template image and search region (using SSD¹ criterion), the system tracks the eye. Two threshold values are calculated in the first frame of this step. By these threshold values, the dissimilarity level in each frame determines three states of the eye: 1) open, 2) closed, and 3) lost. If in several successive frames, the dissimilarity value locates a closed-eye state, a blink is detected. All stages of this step are shown in Figure 1.

Although blink detection can be done in the first step of this experiment, it is carried out in the second step (eye tracking and blink detection), due to the following reasons:

- 1- In the second step, the calculations are done on a limited region surrounding the eyes, but in the first step (eye localization), the calculation includes the whole face and face detection runs every second; therefore, the execution speed of the first step is lower than that of the second step.
- 2- In the second step, error rate is lower than the first step (if the first step is used as a blink detector), since the search region in the second step is smaller and hence the probability of error is lower.

¹Sum of Squared differences

3- In the second step, more information is at hand about eye closure duration.

2.2.1. Eye tracking

At the beginning of the second step, the location of the search region is determined. The initial search region is considered a little larger than the size of open-eye template image; however, the centers exactly fall on each other. In the eye tracking stage, we try to keep the eye (left or right) in the center of the search region. In other words, the search region will be updated in each frame to contain the eye in the new location. To update the search region in each frame, dissimilarity value between the open-eye template and all possible sub-images within the search region (with a size equal to the template) is calculated; according to the minimum value, the new location of the eye is found.

We used SSD function to obtain the dissimilarity value between open-eye template and all possible sub-images within the search region. This function first subtracts all pixels of the first image from those of the second image; then, it squares these differences and the result is the sum of squared differences. The SSD value between the two images (with an equal size) is calculated by:

$$D = \sum (I_1(x, y) - I_2(x, y))^2 \quad (9)$$

where D is the dissimilarity value, I_1 and I_2 are the two equal-size images and x, y are pixel position.

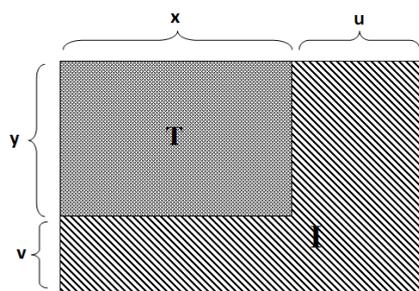


Figure 7. Geometrical description of SSD parameters

Although previous studies [3, 4] have applied normalized correlation coefficient (NCC) for the search template image in the search region, SSD has been used in our proposed system for the sake of fewer computations.

In order to find the most similar sub-image (within the search region) to the open-eye template image, for all possible sub-images (e.g., $u \times v$ as shown in Figure 7), the SSD value is calculated by:

$$D(u, v) = \sum (I(x + u, y + v) - T(x, y))^2 \quad (10)$$

where $I(x, y)$ and $T(x, y)$ are intensities of the pixels at position (x, y) in the search region and template image, respectively; u and v are the movement freedom of the sub-image in the search region. $D(u, v)$ shows dissimilarity intensity between the template image and a sub-image with its top left corner in (u, v) point of the source image. A geometrical description of this stage is shown in Figure 7.

The more the open-eye template and the best match of the search region are dissimilar, the higher D will be. Three examples of eye tracking in the search region are depicted in Figure 8. As it can be seen, the lowest level of dissimilarity between the open-eye template and sub-images in the search region belongs to the one in which the eye is open (Figure 8, left). However, in the sub-image in which the eye is closed, dissimilarity value is neither high nor low; in the sub-image in which the eye is lost, dissimilarity value is high.

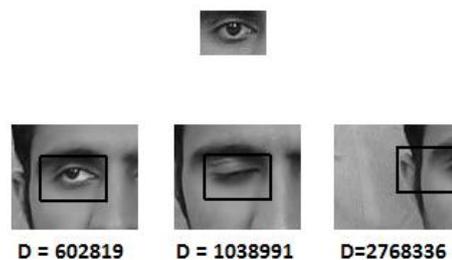


Figure 8. Eye tracking in the search region; (top) open-eye template; (bottom) location of the best match between the template and sub-images in three states: open, closed, and lost in the search region

2.2.2. Blink Detection Using Adaptive Thresholds

For blink detection, our system in each frame classifies the state of the eye as 1) open, 2) closed, or 3) lost, based on dissimilarity value. For this purpose, we have used two thresholds, named open-eye threshold (Th_{open}) and closed-eye threshold (Th_{close}); the open-eye threshold is

lower than the closed-eye threshold. In each frame, one of these three states occurs:

1-Dissimilarity value is less than the open-eye threshold ($D < Th_{open}$): In this state, dissimilarity between the open-eye template and the most similar sub-image in the search region is insignificant; in other words, they are highly similar, thus, the eye is open.

2-Dissimilarity value is greater than the open-eye threshold and less than the closed-eye threshold ($Th_{open} < D < Th_{close}$): In this state, the open-eye template and the most similar sub-image in the search region are almost similar; therefore, the eye is closed.

3-Dissimilarity value is greater than the closed-eye threshold ($Th_{close} < D$): In this state, the open-eye template is not similar to the most similar sub-image in the search region; therefore, it is concluded that the tracker has lost its target (eye). After losing the location of the eye, the first step of our system restarts to locate the eyes and generates a new open-eye template (Figure 1).

By drawing dissimilarity values in successive frames, a curve as in Figure 9 is drawn in which the horizontal axis denotes the frame number and the vertical axis indicates the dissimilarity value.

There are two approaches for defining open-eye and closed-eye thresholds. In the first approach, two constant values are assigned to these thresholds, similar to Grauman's studies [3, 4]. In the second used approach, the thresholds are calculated, based on factors such as illumination, distance between the subject's face and the camera, color of the face, and glasses or beard on the face.

The advantages of adaptive thresholds are listed below:

1- Since SSD-based dissimilarity value is not within a limited range in various conditions (such as illumination changes), the constant threshold values do not work well for our system.

2- Eye closure intensity at a blinking moment varies in different subjects, i.e., a person may almost blink, while another person completely blinks. Hence, SSD value in these conditions is not equivalent.

3- For open-eye template generation, based on the new lighting circumstances, the thresholds are calculated. Therefore, lighting variations do not affect the performance of this adaptive threshold-based system.

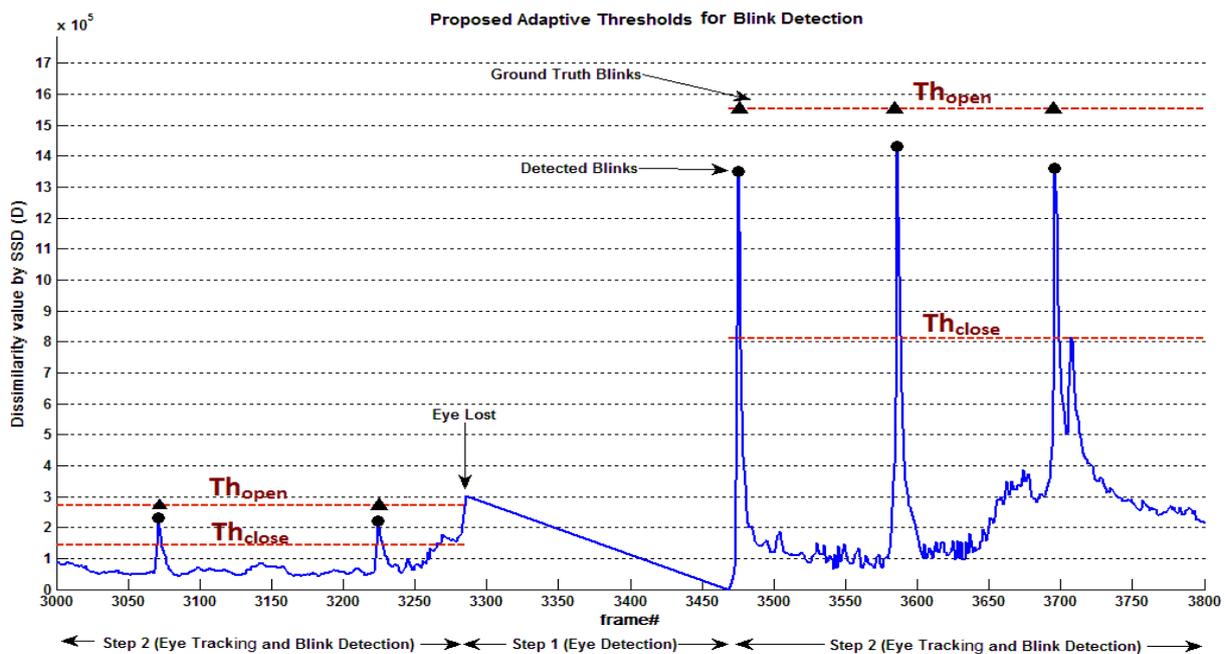


Figure 9. Dissimilarity curve; the triangles are real blink occurrences and the circles on the peaks are the detected blinks by our system. Open-eye and closed-eye thresholds are shown by dashed lines.

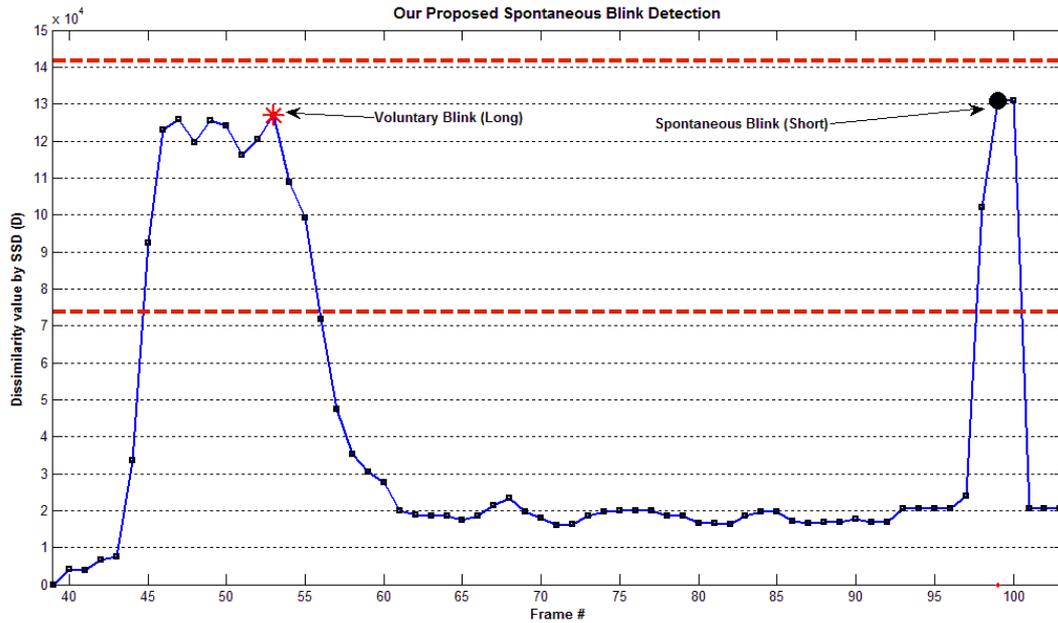


Figure 10. The difference between voluntary and spontaneous blinks on the dissimilarity curve

4- Since the distance between the individual and the webcam leads to various face image sizes, different SSD values are obtained in different distances. Therefore, by using the adaptive thresholds, the subject's distance from the webcam and his/her forward or backward movements from the camera do not affect the overall system performance.

For calculating open-eye and closed-eye thresholds, we proposed a parameter, namely initial open-closed eye dissimilarity, which shows how much the open-eye template is similar to the closed eye in terms of SSD value in the first frame of the second step:

$$D_0 = \sum_{x,y} (I(x,y) - T(x,y))^2 \quad (11)$$

where D_0 is the initial open-closed dissimilarity value, I is the closed-eye image within the search region (five frames after the frame of open-eye template generation), and T is the open-eye template. Note that the sizes of I and T images are equal.

Based on the initial open-closed dissimilarity, our system calculates two thresholds, using the following equations:

$$Th_{Open} = 0.6 * D_0 \quad (12)$$

$$Th_{Close} = 1.15 * D_0 \quad (13)$$

where Th_{Open} is the open-eye threshold and Th_{Close} is the closed-eye threshold.

A dissimilarity curve is drawn in Figure 9, which contains 800 frames (about 26.6 seconds). As it can be seen, in frames 3000-3290, the second step (eye tracking and blink detection) is performed. In frame 3290, the curve exceeds the closed-eye threshold and the system informs us that the eye is lost. Then, the first step (eye localization) starts again. In frame 3480, with the first blink occurrence, the open-eye template image is recreated and the second step restarts. By restarting the second step (frame 3480), the subject's distance from the camera is changed and hence, the thresholds are created, based on the new conditions.

2.2.3. Ignoring long blink durations

Since the goal of this study was the detection of spontaneous blinks, voluntary blinks longer than the spontaneous ones, were ignored. We calculated the average number of frames, in which the eye is not fully open, which is between three and six frames (in a 30 fps video). Hence, we considered closed eyes as blinks if only they were closed in 10 successive frames, at most. In other words, if dissimilarity curve remained more than 10 frames between the two thresholds, no blink was detected.

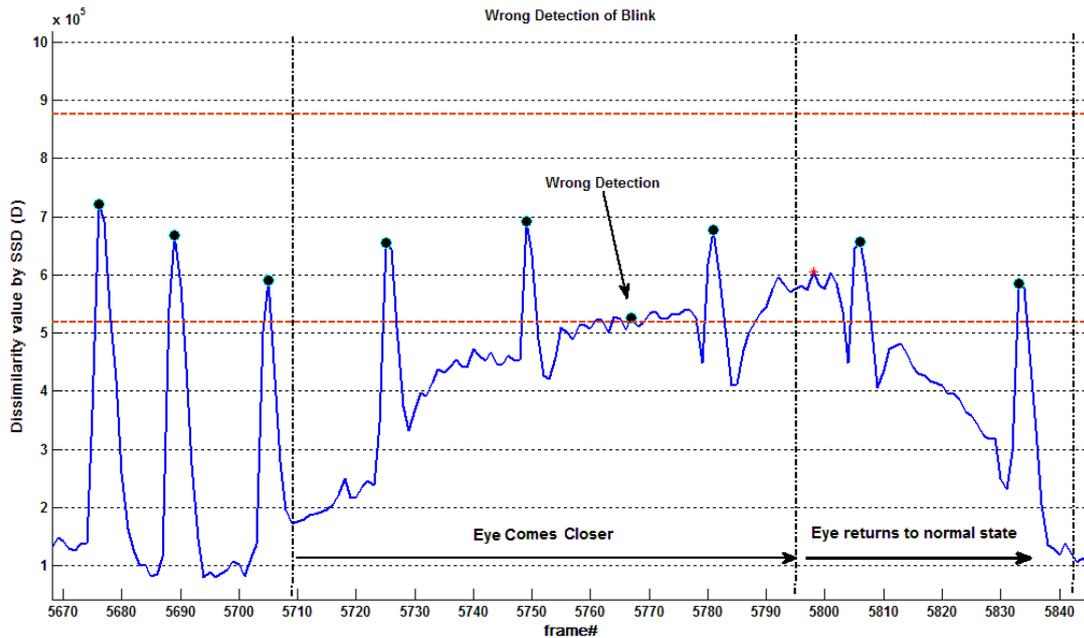


Figure 11. Incorrect blink detection caused by the curve proximity to the open-eye threshold

Figure 10 shows a dissimilarity curve of a voluntary (longer than the usual blink) and a spontaneous blink (shorter than 10 frames). As it can be seen in Figure 10, by the first entrance of dissimilarity curve to the closed-eye region (in frame 46), the curve remains at frame 10 and hence, the system detects and ignores the voluntary blink. However, in frame 98, the curve stays at frame 3; thus, the system considers it as a spontaneous blink. The triangle shows the occurrence of a real blink and the circle depicts the detected blink; the star is a long blink, which the system ignores.

2.2.4. The Proposed Feature: Peak-To-Neighbors Ratio

One problem, which may persist in the second step of our blink detection system, is the proximity of dissimilarity curve to the open-eye threshold, without eye blinking. With the proximity of dissimilarity curve to the open-eye threshold, any swinging behavior of the curve leads to its entrance to the closed-eye region (between the open-eye and closed-eye thresholds) and our system incorrectly detects this entrance as a blink. The main reason causing this problem is the slow and nominal change of subject's distance from the camera.

At the blinking moment, the change in dissimilarity curve is sudden and salient. Therefore, we introduced a new feature named "peak-to-neighbors ratio". When the value of this feature is higher, a true blink is detected and if the feature's value is insignificant, the system ignores the blink. For more understanding, an example is shown in Figure 11, which shows one wrongly detected blink among eight correctly detected ones.

Note that these incorrect detections increase the FPR of blink detection system and hence, it decreases the system's overall accuracy. For solving this problem, our proposed feature is the ratio of peak height at blinking time (t) to the average of dissimilarity value changes in the last 10 frames, obtained via the following equations: First, the average of dissimilarity values in the last 10 frames is calculated:

$$m_{score}(t) = \frac{1}{10} \sum_{i=t-11}^{t-2} d(i) \quad (14)$$

where $m_{score}(t)$ is the average dissimilarity value of the last 10 frames in frame t , and $d(i)$ is the dissimilarity value in frame i .

Then, the relative height of the dissimilarity value of the blinking frame is calculated by:

$$h(t) = d(t) - m_{score}(t) \quad (15)$$

where $h(t)$ is the relative height of the curve in frame t .

Then, the average of dissimilarity value changes in the last 10 frames is calculated by:

$$m_{\text{change}}(t) = \frac{1}{10} \sum_{i=t-12}^{t-2} (d(i) - d(i - 1)) \quad (16)$$

where $m_{\text{change}}(t)$ is the average of dissimilarity changes in the last 10 frames.

Finally, the final feature value in frame t is calculated by:

$$f(t) = \frac{h(t)}{m_{\text{change}}(t)} \quad (17)$$

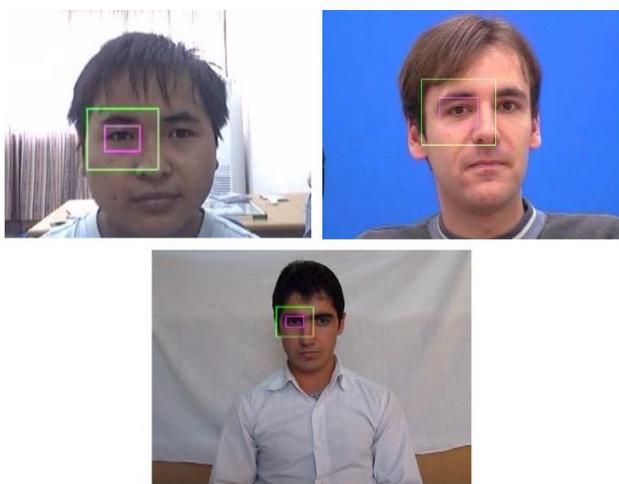


Figure 12. Examples of individuals in three used datasets: (top left) ZJU dataset, (top right) Talking Face dataset, and (bottom) our SBD dataset

3. Results

In order to measure the accuracy of our proposed blink detector, a program was written in MATLAB software (2009) and implemented on a Core2Dou 2.0GHz Intel CPU with 2GB RAM, running on a Windows 7 operating system.

The proposed system of eye localization and blink detection was tested on three video datasets. The first dataset is a well-known blink dataset, named ZJU dataset [10]. This dataset consists of 80 gray-scale videos of 20 individuals (duration of 260 seconds and 255 blinks, in total). The second video dataset is Talking Face Video, which contains a one-person colored video, with

the duration of 166 seconds including 63 blinks [20]. The third one is our generated dataset, named SBD, which contains colored videos of 59 individuals (9,353 seconds and 1303 blinks, in total). All the three dataset videos were in 320×240 pixels. Some video snapshots of these three datasets are shown in Figure 12.

In order to create our SBD dataset, we used a Canon digital camera (PowerShot A710 IS) to capture videos of subjects at the rate of 30 Frame Per Second (fps). The video resolution was originally 640×480, but for testing our method, we reduced it to 320×240 pixels. The captured videos were colored with a pixel depth of 24 bits. The dataset obtained for the evaluation contained videos of people with different facial features (all the subjects were standing in these videos). All participants were males, with or without glasses, and some of them were bearded. The subjects were asked to move as they would normally do when reading a text on a monitor. The light source was different for each individual.

3.1.1. Blink detection results

Some results of the first step (open-eye template images) are presented in Figure 6. Note that a person may have several open-eye templates in a video sequence at different times, since after the eye is lost in the second step, the first step recreates an open-eye template image. In addition, Figure 12 shows three different individuals in three datasets. In this figure, you can see the search region and open-eye template in a rectangle.

Instead of using fixed thresholds, we used adaptive ones for detecting the blinks. Here, for showing the drawbacks of constant open-eye/closed-eye thresholds in contrast with the adaptive ones, we implemented the Grauman's blink detection method [3]. Grauman used NCC for dissimilarity evaluation. In the mentioned blink detection system [3], open-eye threshold is fixed and it is considered to be 0.55; closed-eye threshold is considered to be 0.8.

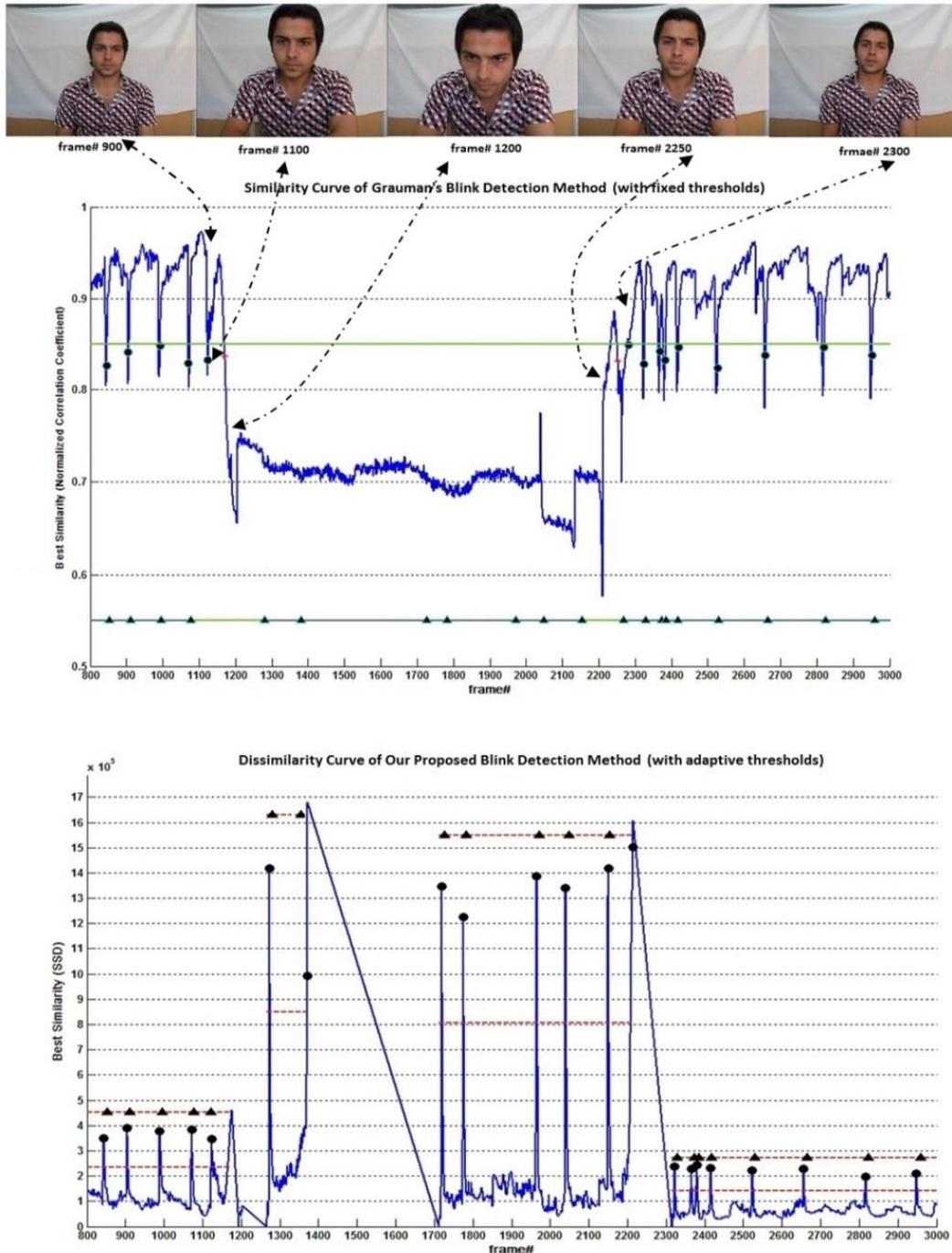


Figure 13. Constant threshold drawback; (Top) subject's images with different distances from the camera; (Middle) similarity curve obtained by Grauman's method [3]; (Bottom) dissimilarity curve obtained by our system. The triangles are real blinks (Ground Truth) and the circles are the detected blinks.

For comparing our proposed system with Grauman's [3], we chose a video from SBD dataset and applied both approaches. Dissimilarity curve of the proposed system and similarity curve of Grauman's are shown in Figure 13 (middle and bottom). Note that the dissimilarity curve of our system behaves as

upward peaks at blinking instances, as shown in Figure 13 (middle); on the other hand, the similarity curve of Grauman's system behaves as downward peaks at blinking instances, indicated in Figure 13 (bottom). The thresholds are drawn as two parallel horizontal lines. In Figure 13 (bottom), given the use of adaptive thresholds, these two lines vary according to

different conditions; however, in Figure 13 (middle), these two lines are continuous, considering the application of constant thresholds.

Five images of a subject at different moments are shown in Figure 13 (top), in which the subject's distance from the camera is normal, a little near, very near, a little near, and normal again (from left to right). The subject is near a little in frame 1100 and comes very near in frame 1200; therefore, the similarity curve of Grauman's system [3] enters the range of two thresholds and remains in this range until frame 2250. The reason is that the subject has smoothly changed his distance from the camera and the similarity value is neither high nor low. However, in our proposed system, this problem is corrected by adaptive thresholds (see Figure 13, bottom). As it is indicated, the eye is lost in frame 1200 and the first step starts again with recreating the open-eye template.

3.1.2. Blink Detection Performance

The results showed that our system could operate with an overall accuracy of 96.03%, 87.12%, and 98.59% on ZJU, Talking Face Video, and SBD datasets, respectively.

		Real state	
		Blink	No blink
Detected	Blink	TP	FP
	No blink	FN	TN

Figure 14. Terms used in performance analysis

To evaluate our system, we evaluated performance with three measures. These measures are precision, sensitivity, and accuracy. Our blink detection measures are listed in Table 1. These measures were calculated using equations (18), (19), and (20):

$$\text{Precision} = \frac{TP}{TP + FP} \quad (18)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (19)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

TP, TN, FP, and FN are defined in Figure 14.

Table 1. Performance of the current blink detection system, tested on three video datasets

	Dataset		
	SBD	ZJU	Talking Face
Precision	87.52 %	94.8 %	94.11 %
Sensitivity	81.27 %	92.94 %	87.12 %
Overall accuracy	98.59 %	96.03 %	97.99 %

In Table 2, the accuracy of our system and other blink detection systems is presented. Since Grauman's system [3] is a well-known approach for blink detection and it is cited in several papers, we chose it as an approach for comparison. Also, Divjak's blink detection system [11] and some other blink detection systems are listed in Table 3.

3.1.3. Blink detection speed

Our proposed system used SSD value instead of NCC. Note that SSD calculation is simpler than NCC calculation. SSD execution order is $O(n)$, whereas that of NCC is $O(n^2)$. The simplicity of SSD leads to fast eye tracking in the second step; therefore, the overall speed of our proposed system increases.

In Table 3, the speed of the proposed system is presented; speeds of the first step and the second step are also listed. In addition, the average speed of the system is shown in the third row.

As it can be seen in Table 3, other blink detection systems such as Grauman's [3] and Divjak's [11] are also tested on SBD and ZJU datasets and their corresponding speed values are listed. These results are based on different CPU characteristics, mentioned in the caption of Table 3. Compared to other approaches, the speed of our proposed system, after being tested on ZJU dataset, was higher; the frame rate of the current method reached 35.04 fps.

4. Discussion and Conclusion

The algorithm presented in this paper can detect the face, track the eyes, and detect blinks at 35 fps in a medium-level personal computer, with 98.59% accuracy.

The first innovation of this system is that conditions are not limited. In other words, lateral movements of the subject, severe lighting changes, diversity of subjects' characteristics (e.g., gender, skin color, eye color, eye shape, eye size, and blinking behaviors), different orientations of the head, and presence of distracting objects on the face (e.g., glasses and beard) do not affect the accuracy of our blink detection system. This is

due to the application of adaptive thresholds for classifying eyes as open or closed.

The second advantage of the proposed system is the increased processing rate (35 fps). This speed enhancement is related to the use of SSD instead of NCC for tracking the eyes (SSD and NCC time complexities are $O(n)$, and $O(n^2)$, respectively). In addition, a new feature named "peak-to-neighbors ratio" was proposed in the blink detection stage to enhance the overall accuracy of blink detection.

For future studies, it can be used both eyes to achieve higher accuracy.

Table 2. Comparison between the overall accuracy of our method and other approaches, tested on ZJU, SBD, and Talking Face datasets

	The proposed method	Grauman[3]	Divjak [11]	Danisman [14]	Torricelli [13]	Pan [10]	HMM [10]	Cas- Adaboost [10]
ZJU	96.03	92.95	97±7	94.8	95.7	95.7	63.4	78.1
Talking Face	87.12	83.45	88	----	----	----	----	----
SBD	98.59	95.33	----	----	----	----	----	----

Table 3. Speed of the proposed blink detection approach in comparison with that of other blink detection methods (in terms of fps)

	The proposed method [†]		Grauman [3] [†] (implemented in this study)		Torricelli [13] [†]	Divjak [11]	Pan [10] ¹
	ZJU	SBD	ZJU	SBD	ZJU	ZJU	ZJU
Step 1: Eye localization	17.39	23.49	----	----	----	----	----
Step 2: Eye tracking and blink detection	41.58	21.41	----	----	----	----	----
Fps average	35.04	22.28	8.92	8.51	30	20 - 30	20

¹:CPU 2.0 GHz, RAM 1 GB

[†]:CPU 2.0 GHz, RAM 2 GB

ZJU: size 320×240, with a total of 12,000 frames

Acknowledgment

The authors thank and appreciate all persons those help us to generate our video dataset and also Computer Eng. Department of Ferdowsi University of Mashhad.

References

1. Patel KG. Eyelid force measurements and electromagnetism blink stimulation system: Toronto; 2006.
2. Castro FL. Class I infrared eye blinking detector. *Sensors and Actuators*. 2008;A 148:388–94.
3. Grauman K, Betke M, Lombardi J, Gips J, Bradski GR. Communication via eye blinks and eyebrow raises: video-based human-computer interfaces. *Universal Access in the Information Society*. 2003;2(4):359-73.
4. Grauman K, Betke M, Gips J, Bradski GR, editors. Communication via Eye Blinks - Detection and Duration Analysis in Real Time. *IEEE Conference on Computer Vision and Pattern Recognition*; 2001.
5. Królak A, Strumiłło P. Eye-blink detection system for human-computer interaction. *Universal Access in the Information Society*. 2012;11(4):11.
6. Morris T, Blenkhorn P, Zaidi F. Blink detection for real-time eye tracking. *Journal of Network and Computer Applications*. 2002;25(2):129-43.
7. Tian YI, Kanade T, Cohn J, editors. Dual-State Parametric Eye tracking. *Fourth IEEE International Conference on Automatic Face and Gesture Recognition*; 2000 March; Grenoble.
8. Heishman R, Duric Z. Using Image Flow to Detect Eye Blinks in Color Videos. *IEEE Workshop on Applications of Computer Vision (WACV'07)*2007.
9. Bhaskar TN, Keat FT, Ranganath S, Venkatesh YV. Blink Detection and Eye Tracking for Eye Localization. 2003.
10. Pan G, Sun L, Wu Z, Lao S. Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcam. *IEEE International Conference on Computer Vision (ICC'07)*; October; Rio de Janeiro, Brazil2007. p. 14-20.
11. Divjak M, Bischof H. Eye blink based fatigue detection for prevention of Computer Vision Syndrome. *IAPR Conference on Machine Vision Applications*; May 20-22; Yokohama, Japan2009.
12. Blehm C, Vishnu S, Khattak A, Mitra S, Yee RW. Computer Vision Syndrome: A Review. *Survey of ophthalmology*. 2005;50(3).
13. Torricelli D, Goffredo M, Conforto S, Schmid M. An adaptive blink detector to initialize and update a view-based remote eye gaze tracking system in a natural scenario. *Pattern Recognition Letters*. 2009;30:1144-50.
14. Danisman TL, Univ. Lille 1, Lille, France, Bilasco IMD, C.; Ihaddadene, N. . Drowsy Driver Detection System Using Eye Blink Patterns. *International Conference on Machine and Web Intelligence (ICMWD)*; 3-5 Oct. 20102010. p. 230 - 3
15. Kim D, Choi S, Choi J, Shin H, Sohn K, editors. Visual fatigue monitoring system based on eye-movement and eye-blink detection *Stereoscopic Displays and Applications XXII*; 2011 February 11, 2011.
16. Xu Y, Jiang Y, Sun Y. Blink Detection Using 3D Cross Model. *Computational Intelligence and Design (ISCID)*; 28-29 Oct. 2012; Hangzhou2012. p. 115 - 8
17. Khan M, Raza M. Suitable Length of Text Line on the Bases of Eye Blink for Reducing Maximum Focus Losses. *International Journal of Computer Applications (IJCA)*. 2012;30(8):7.
18. Viola P, Jones M. Robust real-time face detection. *International Journal of Computer Vision*. 2004;57(2):137–54.
19. ImranKhan M, BinMansoor A. Real time eyes tracking and classification for driver fatigue detection. *ICIAR2008*. p. 729-38.
20. Talking Face Video [Internet]. *Face&Gesture Recognition Working group*. 2000.