International Journal of Advanced Computer Research
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
Volume-5 Issue-19 June-2015

A Fast and Adaptive Video-Based Method for Eye Blink Rate Estimation

Gheis Mohammadi[1]*, Jamshid Shanbehzadeh[2] and Abdolhossein Sarrafzadeh[3]
Department of Computer Engineering, Science and Research branch, Islamic Azad University of Tehran, Iran[1]
Department of Computer Engineering, Tarbiat Moallem University of Tehran, Iran[2]
Department of Computing at Unitec, New Zealand[3]

Received: 16-May-2015; Revised: 15-June-2015; Accepted: 16-June-2015

Abstract

Eye blink rate (EBR) estimation is one of the informative cues and challengeable areas in eye-based systems that has a wide range of applications like the detecting a driver's drowsiness, anxiety analysis, diseases detection and etc. This paper presents an adaptive blink rate estimation algorithm. The advantages of this algorithm are simplicity, accuracy, fastness, low computational cost and robustness against lighting conditions. This algorithm is based on simple image processing techniques. The first step of blink detection method is eye detection. To accomplish this task, we suppose that a fairly large face image is available. Each frame of the input video is processed and the location of the eye is found. The next step calculates a value to determine the state of eye. Our method uses this value to EBR estimation. This paper presets the accuracy of new algorithm by providing a data set of several people and comparing the results with some of the strong relevant methods. The experimental results show that the proposed method has overall accuracy of 98.91%. The average blink rate estimation time of new algorithm for a sample is less than 80 micro seconds, which makes it suitable for real-time applications.

Keywords

Eye blink rate estimation, Video based, Machine Vision, Human computer interaction, Low resolution camera, Adaptive method.

1. Introduction

The eye has a wide range of applications in computer science area. Eye blink rate (EBR) estimation is one of the informative cues and challengeable areas in computer vision. EBR or the frequency of eye blinks can be obtained by counting the number of eye blinks per minute, which can be measured using facial electrodes [1] or a video camera. EBR estimation has a wide range of applications such as detection of physiological or psychological conditions of human disease [2], detecting a driver's drowsiness [3] [4], health recommendation systems [5], depression detection [6], stress detection [7], predicting reinforcement learning [8] and otherwise. In addition to these applications, EBR estimation has potential to be used in many other applications, such as aliveness and alertness detection, mental focus estimation, teacher effectiveness, lie detection and anxiety analysis.

Several eye state recognition methods have been developed recently. *Although* these methods appear promising in the field of eye state recognition, the eye blinking is very fast and these methods wouldn't be useful for blink rate estimation due to the rigid nature of computing. Beside, many of these methods require high quality images and are sensitive with respect to illumination changes. This paper presents a low computational method based on an innovative method for blink detection [9] that uses a simple low resolution webcam. It requires no offline training and there is no dependency on locating specific eye features. Instead, it is based on general appearance of eye. This method is fast enough and can detect blink rates in variable lighting conditions.

The organization of the rest of this paper is as follows. Section II examines related works. Section III presents the proposed algorithm. Section IV

*Author for correspondence

This work was supported in part by the Department of Computer Engineering, Science and Research branch, Islamic Azad University of Tehran, Iran.

outlines the experimental results and Section V presents the conclusion.

2. Related Works

In comparison to head-mounted eye blink detection, remote detection has several advantages such as ease of use and non-invasiveness. A system that can remotely detect blinks under varying light conditions without additional hardware such as infrared lamps is more healthy and suitable for a wide range of applications. This system should be capable of operating with low-resolution cameras under typical indoor environments. There are several eye tracking systems that can track eyes under these conditions but most of them are unsuitable for EBR estimation. Some eye blinks detection methods are presented next.

Morris et al. [10] introduced a real time blink detection algorithm based on the calculation of a variance map and analysis of eye corners. The precision of this method is about 95% True Positive.
Sirohey et al. [11] proposed an eye blink detection algorithm by localizing eye features and analyzing their movements. It uses normal flow to estimate the motion information. They simulated head movement by using an affine model based on head motion to extract eye. This algorithm can track iris and eyelid movements with more than 90% accuracy and it is suitable for off-line applications. Later, they introduced a new method that used a deterministic finite state machine (DFSM) with three states to detect eye blink [12].

Chau and Betke [13] proposed an eye blink detection algorithm with overall precision of more than 95%. Their method used correlation with an open eye template and operated in real-time. This algorithm automatically adapted to large head movements. Pan et al. [14] presented a real time method to detect the degree of eye closure. It used a boosted classifier and the models for changing eye states by Hidden Markov Model (HMM). It worked well with common low resolution webcams and provided more than 96% accuracy.

Orozco et al. [15] described a system that used two appearance-based trackers, one for iris tracking and another to detect eyelids and blinking. This system is based on a simple appearance model and can operate in real time using low resolution camera images. Although the accuracy of this method in iris tracking and detecting eyelids is reasonable, its performance in blink detection is poor. Bacivarov et al. [16] proposed a method that models properly the eye region for both eyes open and closed by using a statistical Active Appearance (AAM) based technique. It can be adapted to work in real-time.
Divijak et al. [17] proposed a template-based method that uses user's eye dynamics and blink patterns. It can detect common cases of fatigue behavior with performance more than 95%.

Lee et al. [18] developed a new method for blink detection, which maintains its accuracy even if the facial pose changes. Firstly, the face and eye regions are detected. Secondly, the eye "open" and "closed" states are determined. Thirdly, the accuracy of determining the eye state when it is open or closed is increased and finally, the SVM classifier for determining the eye state is adaptively selected according to the facial rotation. The performance of eye-blink detection by this method is about 96%. Naveed et al. [19] presented an efficient eye tracking system having a feature of eye blink detection for controlling an interface. His system is able to track eye movements efficiently and accurately by using the pupil portion and can accurately detect eye blinks whether voluntary and involuntary. The system can track eye portion with the 90% detection accuracy. The system is expanded to work in real time using recorded videos.

Mai K. Galab et al. [20] proposed a non-intrusive system for detecting eye blinks accurately without any restriction on the background. No manual initialization is required and their system automatically classifies the eye as either open or closed at each video frame. There are a few methods directly able to estimate EBR. In most of the described methods, their overall performance is greatly variable respect to the accuracy of feature localization because they detect blinks by localizing eye parts. If we use low quality webcam images as input, achieving good accuracy is difficult. Additionally, real-time limitation prevents us from using sophisticated eye localization techniques.

3. Proposed method

This section presents an accurate adaptive EBR estimation algorithm that can perform its estimation using low resolution webcams with very low computational cost under variable lighting conditions. The first step in analyzing a blink is to locate eyes. To accomplish this, we suppose that a large face image is available. Each frame of the input video is processed and the location of eye is found. Figure 1 shows the block diagram of the method. Our method is in green area of this chart. The first block captures a RGB color image and the second one detects or tracks the face region of the input image. The third block localizes the eye regions in the predetermined region of the detected face area. The final block uses our method and calculates a value to determine the eye blink parameters and then by analyzing it, we can estimate eye blink rate.

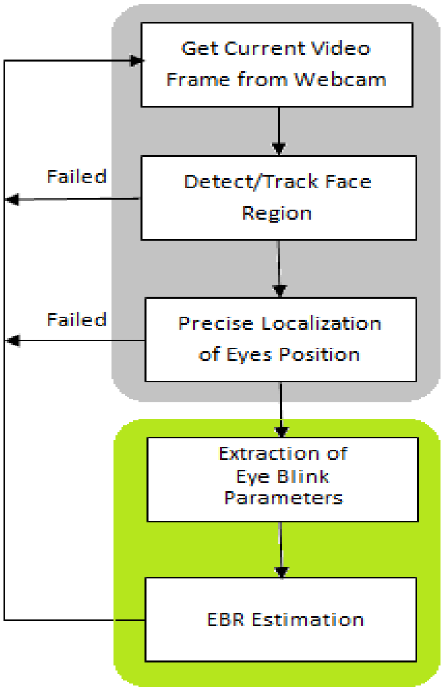

Get Current Video Frame from Webcam
Detect/Track Face Region
Failed
Precise Localization of Eyes Position
Failed
Extraction of Eye Blink Parameters
EBR Estimation

Figure 1: Block diagram of the proposed method

3.1. Detecting face and eye regions

Face and eye regions detection is the first step of the system framework. In the proposed system the Viola Jones algorithm [21] is applied for face detection and tracking. The basic idea of this algorithm is to slide a window across the image and evaluate a face model at every location. In comparison with AdaBoost [22], the Viola Jones algorithm is more efficient for tracking in real-time systems with multiple image frames. This algorithm can detect multi faces and finds target face with the existing of other people or objects. It can track different types of facial views. Viola Jones is characterized by being extremely fast and achieving high detection rates. Each frame of the input video is processed to detect the locations of the user's face and their left and right eye. We use a set of simple geometrical constraints to verify the detected regions. Our method supposes that eyes are located in the second quarter of the head and uses head angle to approximate its location.

3.2. EBR estimation

The first step in the estimation part is to extract eye blink parameters. Figure 2 presents the different states of eye. Any eye state change from open eye to closed and respectively closed eye to open in a limited time can detect as a blink. The states of eye have different appearance properties. If eye is closed, iris and pupil will be invisible, but if it is open, part of iris or pupil will normally be visible. In addition, pupil is the darkest area of eye region, so iris and pupil can provide important information for the states of eye. Using these facts, we can say that if eye is more open then there will be more dark pixels in an eye image. Our method uses this rule to detect eye blink.

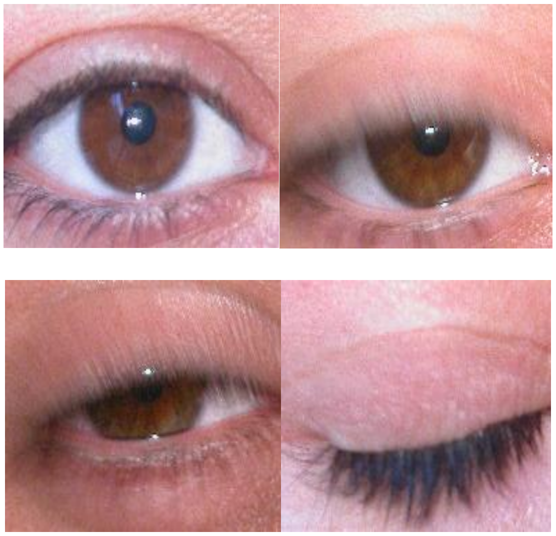


Figure 2: The different states of eye

Our method gives an eye image as input and uses its intensity. The luminance component (Y) of YCbCr color space is chosen as the eye intensity image. The reasons for this selection are as follows:

1. The luminance component of YCbCr is independent of color, so it can be employed

to solve the illumination and eye color variation problem.

2. In comparison with the components of the other color spaces, the luminance component of YCbCr has a relatively higher contrast compared with skin and the white area of the eye.

Therefore, the Y channel of the eye image is used as the intensity image instead of its grey scale version. We use this intensity image to calculate 'τ'. τ is a key intensity level that is changed by eyelid movements. Therefore, it can determine the main states of the eye, open or closed, together with all the intermediate states such as partial blinks, squints, etc.

3.2.1. Computing 'τ'

For computing 'τ' we use a cumulative histogram of the eye intensity image that is calculated by the following steps:

1. Take an eye image.(Figures 3.a.1 and 3.b.1)
2. Extract its luminance component (Y).(Figures 3.a.2 and 3.b.2)
3. Apply a smoothing median filter.(Figures 3.a.3 and 3.b.3)
4. Compute the cumulative histogram 'H' of the filtered image (Figures 3.a.4 and 3.b.4). The cumulative histogram can be found by integrating the histogram of each of the ROI by using (1):

$$H(L) = \sum_{r=0}^{L} h(r) \qquad (1)$$

Where h(r) presents the histogram of the probability of occurrence of intensity level 'r' and 0<r<255.
Figure 3 shows the cumulative histogram of the two main eye states. It can be seen that there is a shift in bars between these two states and this provides a direct relation with the distance of these bars from the origin. Now, we can calculate 'τ' as described as follows:

$$\tau \ = \ r_{max} \ \text{that} \ H(r_{max}) \le T_r \qquad (2)$$

Where τ is the maximum intensity level that its cumulative histogram value is less thanT_r. 'T_r' is a certain threshold value. Tuning 'T_r' can improve execution time and estimation accuracy and robustness, but the sensitivity of the overall system respect to this parameter is low. Figure 4 shows an example of the variations of 'τ_max' respect to Tr for three illumination conditions.

5. $\Delta\tau = \tau(n) - \tau(n-1) \qquad (3)$

Where 'n' is the video frame number.

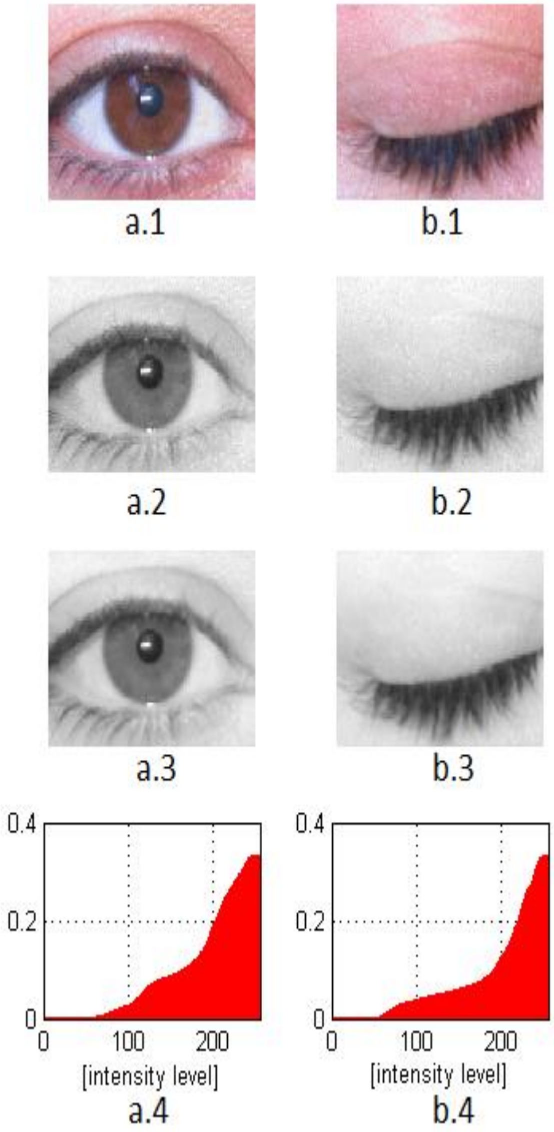


Figure 3: Steps of calculating 'τ' for an open eye (a.1-a.4) and closed eye (b.1-b.4)

We use '$\Delta\tau$' to measure the distance between different states of the eye. The change of eye state produces a mutation in'$\Delta\tau$'. Figure 5 shows an example of 'τ' and '$\Delta\tau$' graphs for a participant left eye that includes slow blinks. In the graphs of'$\Delta\tau$', we can see that there are some points with maximum value. The maximum values appear in the frames in which the subject changes the state of his/her left eye. Therefore, we can use $'\Delta\tau'$ to detect blinks directly without need to detect any extra feature.

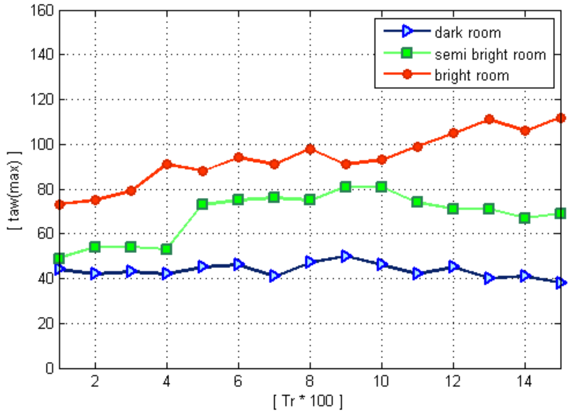


Figure 4: An example of the variations of $'\tau_{max}'$ respect to T_r for three illumination conditions.

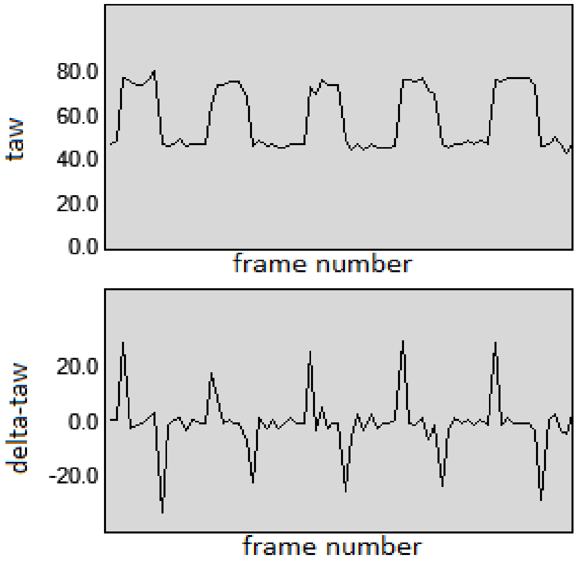


Figure 5: Example of $'\tau'$ and $'\Delta\tau'$ graphs for a participant eye

3.2.2. Computing 'T_d'

The application of $'\Delta\tau'$, in the states of the EBR estimation needs a solution to find a threshold value to recognize and count needle pulses of $'\Delta\tau'$. In different illumination, $'\Delta\tau_{\max}'$ has different magnitude and when the light of the room is higher, $'\Delta\tau_{\max}'$ variations are bigger. Figure 6 shows an example of these variations of $'\Delta\tau_{max}'$ with respect to T_r for three light conditions.

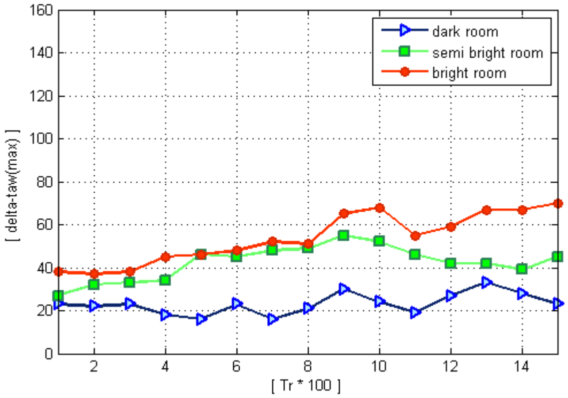


Figure 6: Variations of $'\Delta\tau_{max}'$ respect to T_r for three light conditions

Table 1:Threshold Ranges that are Achieved in our Experiments

Light intensity	Mean eye image intensity	magnitude peaks	T_{min}	T_{max}	T_d
Dark	51.75	[10,19]	3	9	6
	61.66	[12,24]	6	11	9
	65.81	[15,18]	7	14	11
	74.77	[17,24]	6	16	12
	83.06	[21,27]	8	20	14
	102.72	[20,28]	11	19	15
	123.91	[22,34]	12	21	17
	131.81	[22,40]	10	21	16
	141.30	[28,38]	11	27	19
Bright	167.19	[30,44]	9	29	19

With respect to these variations, to detect blink needle pulses that are shown in fig 5, we need an appropriate threshold value for magnitude in every illumination condition. This threshold filters small needle pulses. Our method must find this threshold automatically and without any reconfiguration. To accomplish this, we use a relation between the amount of the illumination and the threshold value. As it's shown in fig 7, in every situation, the mean of intensity is different. The desired threshold value, T_d, is calculated by the use of the following formula:

$$T_d = \left[\frac{T_{min}+T_{max}}{2}\right] + 1 \qquad (4)$$

Where, T_d is the minimum pulse strength for filtering small needle pulses can help us to detect changes of the eye states in the current frame. T_{min} is chosen in such a way that it must be bigger than the maximum of the tiny magnitude peaks of $'\Delta\tau'$. T_{max} is one unit smaller than the minimum of the main magnitude peaks of $'\Delta\tau'$. Table 1 shows an example of the variation of the magnitude peaks of $'\Delta\tau'$ and lower and upper boundaries of the proper threshold values, $'T_{min}'$ and $'T_{max}'$.

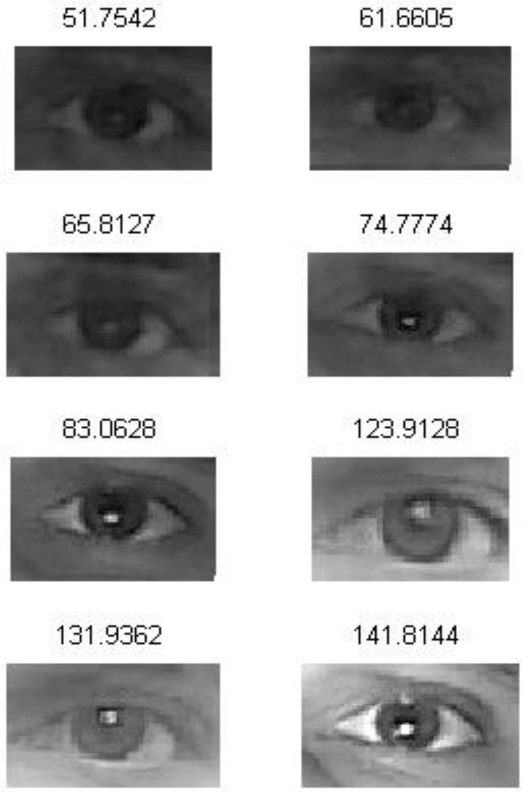


Figure 7: Mean intensities of an eye image in several different illumination conditions

Figure 8 shows threshold values of table 1. We can see the ranges of threshold values and an approximation curve for these thresholds that is drawn in red. As can be seen in Figure 8, we can approximate desired threshold value by an exponential function presented by (5):

$$T_d \approx T_0 + k\left(1 - e^{-a(\bar{I}-\overline{I_0})}\right) \qquad (5)$$

Where, T_0 is the constant desired threshold for a dark room, $\overline{I_0}$ is the minimum mean intensity of the Y channel and is constant and $\bar{I}$ is the mean intensity of the current frame. '*k*' and '*a*' are constant coefficients. A simple way for computing '*k*' and '*a*' is finding the best threshold value for 3 lighting conditions (bright , medium and dark) to achieve the minimum false detection and computing 'a' and 'k' using codes like Figure 9.

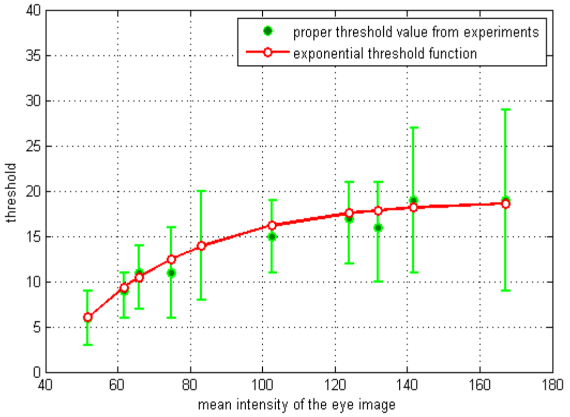


Figure 8: An example of selecting threshold values

```
Algorithm computeCoefficients{
m ← mean intensity of eye image in 3 light condition (1:dark ,
      2:medium , 3:bright)
Ti ← Threshold values for 3 light conditions (1:dark ,
2:medium , 3:bright)
I0 ← m(1): mean intensity of eye image in dark room
desiredError ← desired error (for example 0.5)
k ← Ti(3)-Ti(1)
E ← desiredError

for a = 0 to 10 with step 0.001
  n ← 0
  for i ←  1 to 3
    T(i) ← (k*(1-exp(-a*(m(i)-I0))))+Ti(1)
    e(i)=abs(T(i)-Ti(i))
    if e(i)<= desiredError
      n ← n+1
      if n==3
        tmpE ← E
        E ← (e(2)+e(3))/2
        If E < tmpE
          $a_{desired}$ ← a
        end if
      end if
    else
      break inner loop
    end if
  end for
end for

return k, $a_{desired}$
}
```

Figure 9: Pseudo code for computing '*a*' and '*k*'

In (5), all constant parameters are independent of the hardware. Figure 10 shows the estimated T_d value for a simple video sequence with some blinks.

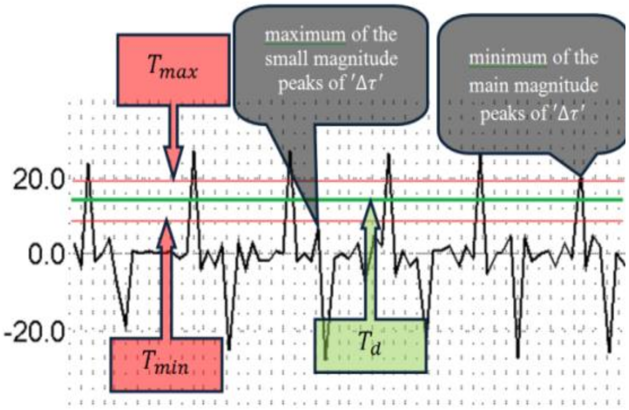


Figure 10: An example of $'\Delta\tau'$ with marked threshold

3.2.3. Counting eye blinks to estimation

After determining the $\Delta\tau$ and T_d, our method uses these values to EBR estimation. Actually, EBR is equal to the number of sequential positive and negative pairs of needle pulses in $\Delta\tau$ that are bigger than T_d.

4. Experimental Results

The hardware of our approach is a webcam connected to a laptop. We employed simple hardware to evaluate the prototype system and to discover how the system can work with low cost components that could be affordable and function well with any other system.

Experimental results were obtained using Microsoft LifeCam VX-700 with resolution of 800 * 600, and a Dell Inspiron 6400 2.0GHz Centrino Dual Core with 2GB RAM laptop. The proposed approach was implemented by Microsoft Visual C#.net using ‘EMGU.CV’ library for image processing. In our experiments, we use k=12 and a=0.05. While preliminary tests have shown that the proposed approach works well in any environment, all the results reported here were obtained indoors without lighting control.

Table 2 shows the average execution time of each part for a single frame using our C#.net program. Table 2 shows that the EBR estimation part is very fast and its execution time is about 80 micro-seconds.

Table 2: Average Execution Time of Each Part of Our Method for a Single Frame

Algorithm Parts	Time(ms)
Face detection (Using Viola Jones algorithm)	17.90
Face tracking	9.42
Eye detection	10.29
Eye tracking	6.83
’τ’ Calculation	0.07
EBR Estimation	0.01
total average execution (Face & Eye Tracking + Blink Detection)	16.33

In another experiment, the performance of the present method is compared with some recent methods and Table 3 presents the results. We can see that the accuracy of our method is higher than other methods and the proposed method is much faster.

Table 3: Comparison of Some Blink Detection Methods with Proposed Algorithm

Method	EBR Estimation accuracy	EBR Estimation Time (milliseconds)
[17]	90.25%	[3.3 – 5]
[20]	98%	7
[23]	91.8%	5
[24]	92.6%	1.14
[25]	94.2 %	15
Ours	98.91%	0.08

In table 3, the performance of our method was measured by performing an experiment with a total of 17 participants (14 males and 3 females) with a wide range of ages and some with individuals wearing glasses. The participants were sitting approximately 70 cm away from the camera. We ask them to fix their heads in the front of camera. They were permitted to blink involuntary or after we ask them to blink. In this experiment, our application captured 18000 frames in 10 minutes from every subject and processed these frames in real time. Table 4 shows the results. The EBR estimation and its error and estimation accuracy are calculated using (6) and (7) and (8), respectively.

$$EBR_{est} = \frac{TP+FP}{t} \tag{6}$$

$$e_{est} = |EBR_{act} - EBR_{est}| \qquad (7)$$

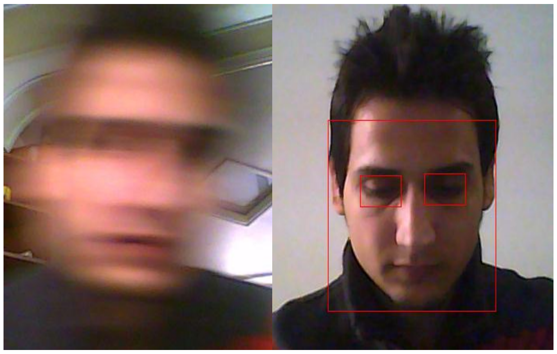
$$Estimation\ Accuracy\ (\%) = 100 \times (1 - \frac{e_{est}}{EBR_{act}}) \qquad (8)$$

Where t is experiment duration in minutes and e_{est} is estimation error and EBR_{act} is user actual EBR and EBR_{est} is estimated EBR. The experimental results show that the proposed method has 98.91% estimation accuracy. From the experiments, two situations occasionally decrease the accuracy of our EBR estimation algorithm. The first one is the result of a swift movement of the user's head (Figure 11.a). In this situation, our algorithm is unable to correctly determine the parameters of eye blink. This is because eye images are blurry so that the skin colors blend with the colors of the eye areas. The second situation is when the user bows his/her head or changes the focus to the lower area (with respect to the camera position) so that the eyelids are captured partially close (Figure 11.b).

Table 4: Blink Detection Accuracy of Our Method

User Id	**Total Blinks (In 5 minutes)**	**Actual EBR (EBR_{act})**	**Blinks**		**Not Blink**	**Estimated EBR (EBR_{est})**	**EBR Estimation Error (e_{est})**	**EBR Estimation Accuracy**
			Number of true needle pulses (TP)	**Number of missed blinks (FN)**	**Number of wrong needle pulses (FP)**			
1	97	9.7	95	2	1	9.6	0.1	98.97%
2	117	11.7	116	1	2	11.8	0.1	99.15%
3	99	9.9	99	0	0	9.9	0	100.00%
4	128	12.8	127	1	2	12.9	0.1	99.22%
5	114	11.4	111	3	0	11.1	0.3	97.37%
6	92	9.2	89	3	2	9.1	0.1	98.91%
7	93	9.3	91	2	1	9.2	0.1	98.92%
8	92	9.2	89	3	2	9.1	0.1	98.91%
9	182	18.2	178	4	1	17.9	0.3	98.35%
10	95	9.5	93	2	0	9.3	0.2	97.89%
11	106	10.6	103	3	1	10.4	0.2	98.11%
12	96	9.6	94	2	1	9.5	0.1	98.96%
13	105	10.5	104	1	0	10.4	0.1	99.05%
14	84	8.4	83	1	1	8.4	0	100.00%
15	81	8.1	79	2	2	8.1	0	100.00%
16	120	12	119	1	0	11.9	0.1	99.17%
17	139	13.9	136	3	1	13.7	0.2	98.56%
Total:	**1840**	**-**	**1806**	**34**	**17**	**-**	**Average: 0.12**	**Average: 98.91%**

TP: Number of blinks that are correctly detected (true positive)
FN: Number of blinks that are missed (false negative)
FP: Number of wrong reported blinks (false positive)



(a) **(b)**

Figure 11: Some detection failed situations
(a) Eye images are blurry such that skin colors blend with the colors of the eye areas
(b) User bows his head and changes the eye focus to the lower area

5. Conclusions

This paper presented a simple real time EBR estimation algorithm that estimates eye blink rate accurately with very low computational cost in real time. The proposed method consisted of these steps: The first step was detecting or tracking the face region in a captured frame. The second step was localizing the eye regions in the predetermined region of the detected face area. In the next step, we calculated a value to determine the eye state and then by analyzing it, EBR were calculated. The proposed approach improved both estimation time and accuracy in comparison with the most relevant methods like the Mai et al method[20] with 98% accuracy or the Matjaž et al method[25] with 94.2% overall accuracy.

6. Future Work

In future, better methods for light invariant ability should be developed according to our presented system. For the next level of improvement some recommendations are given below.

- This system can be implemented by microprocessors and system can be run efficiently.
- The technology of face and eye tracking should be improved for accurate and fast detection.
- This method can be used in many applications like detecting a driver's drowsiness, health recommendation systems, depression detection, stress detection and exc.
- The visual information can be integrated with the proposed system to improve detection accuracy.

References

[1] Robert Krupi´nski, Przemysław Mazurek. "Real-Time Low-Latency Estimation of the Blinking and EOG Signals", Real-Time Systems, Architecture, Scheduling, and Application, Dr. Seyed Morteza Babamir (Ed.), ISBN: 978-953-51-0510-7, InTech, 2012, DOI: 10.5772/38497.

[2] Mitelman R, JoshuaM, Adler A, Bergman H. "A noninvasive, fast and inexpensive tool for the detection of eye open/closed", J Neurosci Methods;2009, 178:350–6.

[3] Jaeik Jo, Sung Joo Lee, Kang Ryoung Park, Ig-Jae Kim, Jaihie Kim, "Detecting driver drowsiness using feature-level fusion and user-specific classification", Expert Systems with Applications 41, 2014, 1139-1152.

[4] Raees Ahmad, J.N. Borole, "Drowsy Driver Identification Using Eye Blink detection", International Journal of Computer Science and Information Technologies, Vol. 6 (1), 2015, 270-274.

[5] Qi Guo, Zixuan Wang, Ming Li and Hamid Aghajan, "Intelligent Health Recommendation System For Computer Users", arXiv: 1504.07858v1 [cs.CV], 29 Apr 2015.

[6] Alghowinem, Sharifa, et al. "Eye movement analysis for depression detection." Image Processing (ICIP), 2013 20th IEEE International Conference on. IEEE, 2013.

[7] Alvaro Marcos-Ramiro, Daniel Pizarro-Perez, Marta Marron-Romera, Daniel Pizarro-Perez, Daniel Gatica-Perez, "Automatic Blinking Detection towards Stress Discovery", Proceedings of the 16th International Conference on Multimodal Interaction, ACM New York, NY, USA, 2014, 307-310.

[8] Heleen A. Slagter, Katerina Georgopoulou, Michael J. Frankb, "Spontaneous eye blink rate predicts learning from negative", but not positive, outcomes, Neuropsychologia, 2015, 0028-3932.

[9] Gheis Mohammadi, Abdolhossein Sarafzadeh, Jmashid ShanbehZadeh, "Real-Time Video-Based Eye Blink Detection", The International Multi Conference of Engineers and Computer Scientists (IMECS), Hong Kong, South Korea, 2011.

[10] T. Morris, P. Blenkhorn, F. Zaidi, "Blink detection for real-time eye tracking, Journal of

Network and Computer Applications", vol. 25, num. 2, April 2002, 129-143.

[11] S. Sirohey, A. Rosenfeld, Z. Duric, "A method of detecting and tracking irises and eyelids in video", Pattern Recognition, vol. 35, num. 6, 2002, 1389-1401.

[12] Ric Heishman, Z. Duric, "Using image flow to detect eye blinks in color videos", Proceedings of the 8th IEEE Workshop on Applications of Computer Vision, 2007, 52.

[13] M. Chau, M. Betke, "Real Time Eye Tracking and Blink Detection with USB Cameras", Boston University Computer Science Technical Report No. 2005-12, May 2005.

[14] Gang Pan, Lin Sun, Zhaohui Wu, Shihong Lao, "Eye blink-based anti-spoofing in face recognition from a generic web camera", The 11th IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.

[15] Javier Orozco, F. Xavier Roca, Jordi Gonzàlez, "Real-time gaze tracking with appearance-based models", Machine Vision Applications, 2008.

[16] I. Bacivarov, M. Ionita, and P. Corcoran, "Statistical Models of Appearance for Eye Tracking and Eye-Blink Detection and Measurement", IEEE Transactions on Consumer Electronics, Vol.54, No.3, 2008, 1312-1320.

[17] Matjaž Divjak, Horst Bischof, "Eye Blink Based Fatigue Detection for Prevention of Computer Vision Syndrome", In Proceedings of the 11th IAPR conference on machine vision Applications, 2009, 10-14.

[18] W. Lee, E. Lee, and K. Park, "Blink Detection Robust to Various Facial Poses", Journal of Neuroscience Methods, 2010, 356–372.

[19] S. Naveed, B. Sikander, and M. Khiya, "Eye Tracking System with Blink Detection", Journal of Computing, Vol.4, No.3, 2012.

[20] Mai K. Galab, H.M.Abdalkader, Hala H. Zayed, "Adaptive Real Time Eye-Blink Detection System", International Journal of Computer Applications (0975 – 8887) Volume 99– No.5, August 2014.

[21] P. Viola and M.J. Jones, "Robust real-time face detection. International Journal of Computer Vision", vol. 57, no. 2, 2004, 137-154.

[22] Lienhart R,Maydt J, "An extended set of haar-like features for rapid object detection", In Proceedings of international conference on image processing, vol. 1, 2002, 900–3.

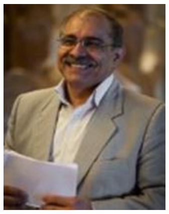
[23] Eric Missimer and Margrit Betke, "Blink and Wink Detection for Mouse Pointer Control", The 3th International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2010.

[24] Ayudhya, Chinnawat Devahasdin Na, and Thitiwan Srinark. "A method for real-time eye blink detection and its application." 6th International Joint Conference on Computer Science and Software Engineering (JCSSE). 2009.

[25] Divjak, Matjaž, and Horst Bischof. "Real-time video-based eye blink analysis for detection of low blink-rate during computer use." First International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS 2008). 2008.

Gheis Mohammadi received his M.Sc. degree in Mechatronic engineering from Science and Research branch of Islamic Azad University of Tehran in 2012 with second rank honor. At present, he is a Ph.D. candidate in the department of electrical and computer engineering, University of Tehran, Iran. His research interests are machine vision, intelligent systems, robotics, human-computer interaction and human-robot interaction.
Email: Gheis.Mohammadi@gmail.com

Dr. Jamshid Shanbehzadeh received his Ph.D. degree in electrical and computer engineering from University of Wollongong (UOW) Institute of Computer, New South Wales, Australia, in 1996. At present, he is an assistant professor in department of computer engineering, Tarbiat Moallem University, Tehran, Iran. His research interests are Image Compression, OCR, Document Analysis, Image Retrieval.



Professor Abdolhossein Sarrafzadeh received his Ph.D. degree in artificial intelligence from University of Wollongong (UOW) Institute of Computer, New South Wales, Australia. Now, he is head of department of computing and director of the Centre of Computational Intelligence for Cyber Security at Unitec in Auckland, New Zealand. His research interests are Artificial Intelligence, Computer Security and Reliability, Human-computer Interaction.