Eye Blink Detection

Patrik POLATSEK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 3, 842 16 Bratislava, Slovakia
patrik.polatsek@gmail.com

Abstract. Nowadays, people spend more time in front of electronic screens like computers, laptops, TV screens, mobile phones or tablets which cause eye blink frequency to decrease. Each blink spreads the tears on the eye cornea to moisture and disinfect the eye. Reduced blink rate causes eye redness and dryness also known as Dry Eye, which belongs to the major symptoms of the Computer Vision Syndrome. The goal of this work is to design eye blink detector which can be used in dry eye prevention system. We have analyzed available techniques for blink detection and designed our own solutions based on histogram backprojection and optical flow methods. We have tested our algorithms on different datasets under various lighting conditions. Inner movement detection method based on optical flow performs better than the histogram based ones. We achieve higher recognition rate and much lower false positive rate than the-state-of-the-art technique presented by Divjak and Bischof.

1 Introduction

The number of people using computers every day increases. There are also more people who suffer from symptoms collectively called *Computer Vision Syndrome* (CVS). It is a set of problems related to computer use. The rate of unconscious eye blinking while looking at luminous objects within close distance reduces significantly (up to 60 % reduction). Blinking helps us to spread the tear film and moisten the surface of the eye, due to which the reduced rate of blinking leads to *Dry Eye*. Typical ocular complaints experienced by intensive computing work (more than 3 hours per day) include dryness, redness, burning, sandy-gritty eye irritation or sensitivity to light and eye fatigue. The easiest way to avoid the symptoms of Dry Eye is to blink regularly [3, 13].

Our aim is to create eye blink detector, which could be used in real-time blink detection system. In case of low blink rate it will notify a user to blink more frequently. This paper proposes two different methods on blink detection. The first of presented algorithms computes *backprojection* from 1D saturation and 2D hue-saturation histogram. The second method addressed as *Inner movement detection* detects eyelid motion using *Lucas-Kanade* (KLT) feature tracker [11].

* Bachelor study programme in field: Informatics
Supervisor: Andrej Fogelton, Institute of Applied Informatics, Faculty of Informatics and Information Technologies STU in Bratislava

IIT.SRC 2013, Bratislava, April 23, 2013, pp. 1–8.

2 Related Work

Optical flow in [7] tracks eyelid movements to detect eye blinks. Detection is based on matching SIFT (scale-invariant feature transform) descriptors computed on GPU. First, thresholded frame difference inside the eye region locates motion regions. Consequently, these regions are being used to calculate the optical flow. While user blinks, eyelids move up and down and the dominant motion is in vertical direction. This method detects 97% of blinks on their dataset. Most of the false positive detections are the result of gaze lowering and vertical head movements. Method based on optical flow estimation is also presented in [4]. It locates eyes and face position by 3 different classifiers. The algorithm is successful mostly when the head is directly facing the camera. The KLT tracker is used to track the detected feature points. This blink detector uses GPU-based optical flow in the face region. The flow within eyes is compensated for the global face movement, normalized and corrected in rotation when eyes are in non-horizontal position. Afterwards dominant orientation of the flow is estimated. The flow data are processed by adaptive threshold to detect eye blinks. Authors report good blink detection rate (more than 90%). However this approach has problems with detecting blinks when eyes are quickly moving up and down.

The eyelid movements are estimated by *normal flow* instead of optical flow in [6]. It is the component of optical flow that is orthogonal to the image gradient. Authors claim that the computation of normal flow is more effective than the previous method.

Arai et al. present Gabor filter-based method for blink detection in [1]. *Gabor filter* is a linear filter for extracting contours within the eye. After applying the filter, the distance between detected top and bottom arc in eye region is measured. Different distance indicates closed or opened eye. The problem of arc extraction arises while the person is looking down.

Variance map specifies distribution of intensities from the mean value in an image sequence. The intensity of pixels located in eye region changes during the blink, which can be used in detection process as in [10].

Correlation measures the similarity between actual eye and open eye image. As someone closes eyes during the blink, correlation coefficient decreases. Blink detection via correlation for immobile people is presented in [5].

A blink detection algorithm in [9] is based on the fact that the *upper* and *lower part* of eye have different distribution of *mean intensities* during open eyes and blinks. These intensities cross during the eyelid closing and opening.

Liting et al. [8] use a deformable model - *Active Shape Model* represented by several landmarks as the eye contour shape. Model learns the appearance around each landmark and fits it in the actual frame to obtain new eye shape. Blinks are detected by the distance measurement between upper and lower eyelid.

Ayudhaya et al. [2] detect blinks by the *eyelid's state detecting* (ESD) *value* calculation. It increases the threshold until the resulting image has at least one black pixel after applying median blur filtering. This threshold value (ESD) differs while user blinks.

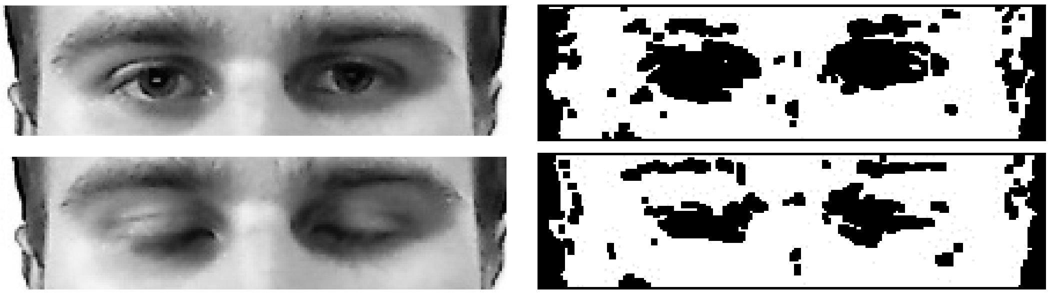
3 Proposed Algorithms

Due to our goal to do CVS preventing system, our main focus is on model situation when the user is facing the computer screen. Because of this, the high recognition rate within the efficient computation is necessary.

We introduce two methods based on histogram backprojection and the Inner movement detection based on KLT feature tracker.

3.1 Histogram Backprojection

We use *histogram* to represent skin color of the user. *Histogram backprojection* creates a probability map over the image. In other words backprojection determines how well the pixels from the image fit the distribution of a given histogram. Higher value in a backprojected image denotes more likely location of the given object. We detect closed eyes as high percentage of skin color pixels within the eye region otherwise we consider eyes opened (Figure 1).



(a) Source images. (b) Backprojected images.

Figure 1. Histogram backprojection for a person with open and closed eyes.

We use the *HSV* (Hue Saturation Value) color model to achieve partial luminance invariance by the omission of the Value channel. We did experiments with two different histograms:

- 1D *saturation* histogram (histogram S),
- 2D *hue-saturation* histogram (histogram HS).

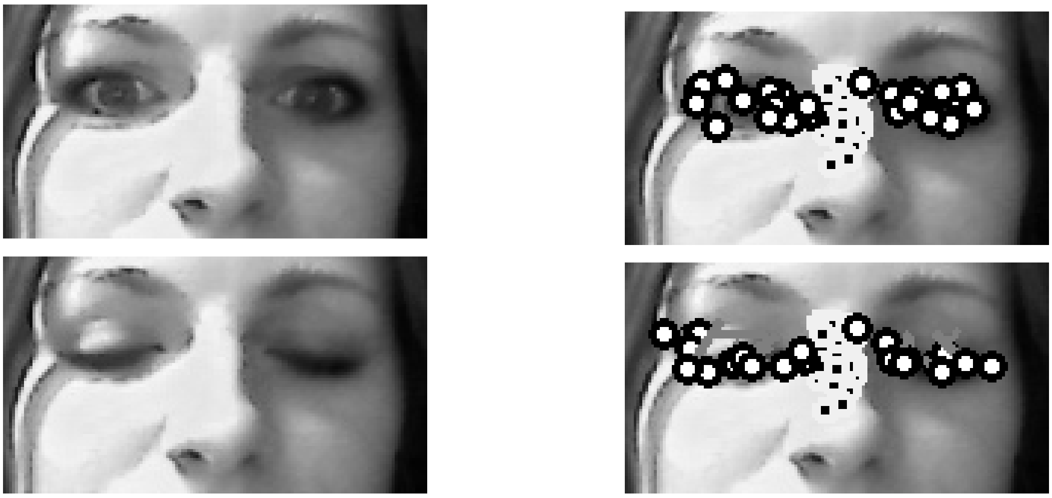
First we detect the user's face by *Haar Cascade Classifier* [12]. We calculate the skin color histogram from a sequence of images of face regions. Other parts of the image are not used to obtain as precise skin color histogram as possible. Histogram is normalized afterwards and regularly updated. For every input image we calculate the backprojection with this histogram. Subsequently a resulting backprojected image is modified using morphological operations (Open and Erode) and threshold ($threshold_{HS} = 10$ in hue-saturation and $threshold_S = 25$ in saturation histogram obtained by experiments) to increase small difference between open and closed eyelids due to lower skin probability of eyelids caused by shadows in eye areas or make-up. Finally the average value of the probabilities is calculated from the region of the user's eyes. Significant increase is considered as eye blink of the user. Figures 3(a) and 3(b) illustrate results of backprojection while using different histograms.

3.2 Inner Movement Detection

We introduce our own *Inner Movement Detection* algorithm based on *optical flow*. Optical flow locates new feature position in the following frame. One of the most common method called KLT tracker [11] selects features suitable for tracking with high intensity changes in both directions.

If a user blinks, the mean displacement of feature points within the eye region should be greater than the displacement of the rest of the points within the face area (Figure 2).

The first step consists of localizing a user's face and eyes using *Haar Cascade Classifier* [12] on grayscale image. We initialize random KLT features within the eye and nose regions and classify them as left ocular, right ocular or non-ocular. These features are being tracked by KLT tracker. Tracking is reinitialized in regular intervals or in case of loss of many feature points. We compute the average displacement separately for three groups of points. Afterwards we compare the difference between the left or right ocular and non-ocular movement displacements. If this difference exceeds a threshold value ($threshold_{diff} = face.height/165$, where $face.height$ is the height of detected face in the initial phase), a movement within the eye region is anticipated. Consequently we count the ratio of ocular points that moved down at least of a specific distance in the direction of y-axis ($distance_y = face.height/110$) in order to exclude false positives caused by horizontal eye movements. Due to proper computation of the ratio we eliminate the vertical ocular displacement caused by head movements. The ocular points are therefore shifted in a distance equal to the average displacement of non-ocular points. If the ratio is higher than a threshold (5% of displacements of one group of ocular points and 2% of displacement of the second one), we consider it as a blink. Figure 3(c) represents a graph of values defined as $max(abs(avg(left) - avg(non)), abs(avg(right) - avg(non)))$, where max and abs are the maximum and absolute value, avg indicates the average movement within a given region and $left$, $right$ and non denote left ocular, right ocular and non-ocular region.



(a) Source images. (b) Displacement of feature points.

Figure 2. Tracking of feature points using KLT feature tracker by eye blink.

4 Evaluation

Our blink detection algorithms are evaluated on two datasets. Our own dataset includes 8 individuals (5 males and 3 females, one person wearing glasses) under different lighting conditions who sit in front of a computer screen mostly in a stable position and looking directly at the screen. It consists of 7569 frames and 128 blinks. The second image sequence - *the Talking Face Video* (TALKING) is publicly available from Inria[1]. It includes 5000 images of a person engaged in conversation who blinks 61 times.

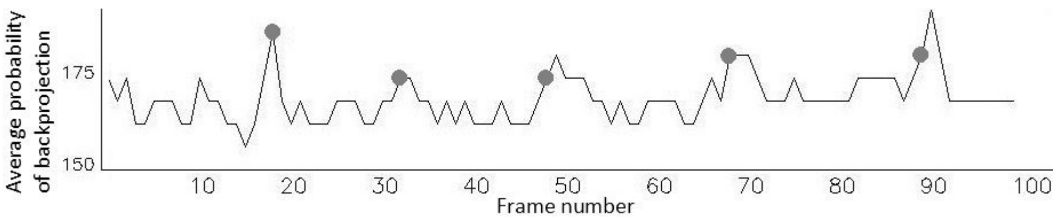
We have tested our algorithms and compared their blink detection abilities to the optical flow method mentioned in [4]. The results are shown in Table 4. The best true and false positive rate are achieved by Inner movement detection. It detects 93,75% of blinks on own dataset and 98,36% of blinks on the Talking Face Video.

Backprojections using hue-saturation and saturation histogram provide similar accuracies. Values of saturation channel of an image differ in case of skin and pupil in most light conditions, thus it provides reliable information about user's blinks. However hue channel is often different in whole eye regions. Sometimes it is without any significant changes when eye blinks. It happens mostly in very dark images. False detections are the results of luminance changes, poor light conditions,

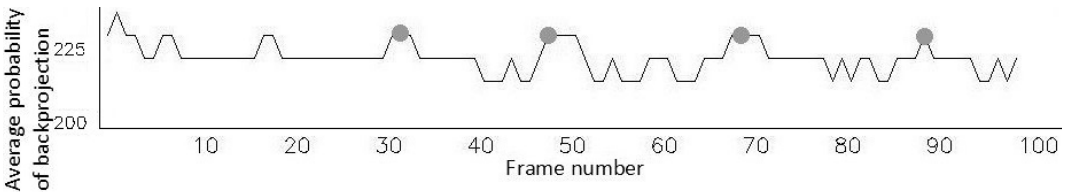
[1] `http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html`

changes in gaze direction, facial mimicry such as smiling and eyelid makeup. In such cases it is very difficult to recognize whether a user blinks or not. Backprojection using hue-saturation histogram has many missed blinks when an individual wears glasses.

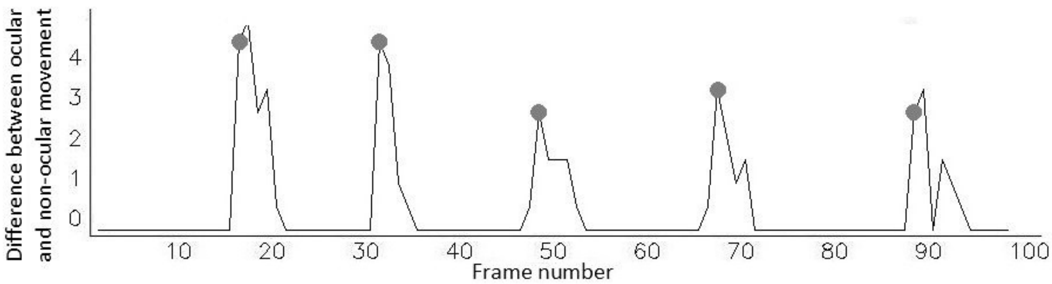
Inner movement detection, our best method, has 14 false positive and 9 false negative cases caused mainly by rapid head movements, lowering the gaze and reflection from glasses.



(a) Graph of the average values of backprojection using 2D hue-saturation histogram.



(b) Graph of the average values of backprojection using using 1D saturation histogram.



(c) Graph of the differences between the average ocular and non-ocular movement within the face area.

Figure 3. Graphs produced using our eye blink detection algorithms. They are computed from an image sequence of a user while working in front of computer. The user blinks at frames 18, 33, 50, 69 and 90. Detected blinks are represented by circles on the graph.

Table 1. Comparison of our blink detection algorithms to the method in [4]. TP represents true positive rate and FP is false positive rate.

Method	Own dataset		TALKING	
	TP	*FP*	*TP*	*FP*
Backprojection (Histogram S)	81,25%	0,40%	88,52%	0,49%
Backprojection (Histogram HS)	75,00%	0,32%	85,25%	0,47%
Inner movement detection	93,75%	0,05%	98,36%	0,20%
Method in [4]	-	-	95%	19%

5 Conclusion

In this paper we proposed two techniques for eye blink detection. The first method detects blinks by backprojection using saturation or hue-saturation histogram. The second method is based on KLT feature tracker, which tracks eyelid motions. The model situation is a user looking at the computer screen. Inner movement detection method outperforms the method in [4]. It provides over 3% better true positive rate and about 18% lower false positive rate.

Acknowledgement: This project is supported by Tatra bank foundation E-Talent 2012et009.

References

[1] Arai, K., Mardiyanto, R.: Comparative Study on Blink Detection and Gaze Estimation Methods for HCI, in Particular, Gabor Filter Utilized Blink Detection Method. In: *Proceedings of the 2011 Eighth International Conference on Information Technology: New Generations*. ITNG '11, Washington, DC, USA, IEEE Computer Society, 2011, pp. 441–446.

[2] Ayudhaya, C., Srinark, T.: A method for a real time eye blink detection and its applications. In: *The 6th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2009, pp. 25 – 30.

[3] Blehm, C., Vishnu, S., Khattak, A., Mitra, S., Yee, R.W.: Computer Vision Syndrome: A Review. *Survey of Ophthalmology*, 2005, vol. 50, no. 3, pp. 253 – 262.

[4] Divjak, M., Bischof, H.: Eye blink based fatigue detection for prevention of Computer Vision Syndrome. In: *IAPR Conference on Machine Vision Applications (MVA 2009)*, 2009, pp. 350–353.

[5] Grauman, K., Betke, M., Gips, J., Bradski, G.: Communication via eye blinks - detection and duration analysis in real time. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. Volume 1., 2001, pp. I–1010 – I–1017 vol.1.

[6] Heishman, R., Duric, Z.: Using Image Flow to Detect Eye Blinks in Color Videos. In: *Applications of Computer Vision, 2007. WACV '07. IEEE Workshop on*, 2007, p. 52.

[7] Lalonde, M., Byrns, D., Gagnon, L., Teasdale, N., Laurendeau, D.: Real-time eye blink detection with GPU-based SIFT tracking. In: *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision*. CRV '07, Washington, DC, USA, IEEE Computer Society, 2007, pp. 481–487.

[8] Liting, W., Xiaoqing, D., Changsong, L., Wang, K.: Eye Blink Detection Based on Eye Contour extraction. In: *Image Processing: Algorithms and Systems*, SPIE Electronics Imaging, 2009, p. 72450.

[9] Moriyama, T., Kanade, T., Cohn, J., Xiao, J., Ambadar, Z., Gao, J., Imanura, M.: Automatic recognition of eye blinking in spontaneously occurring behavior. In: *Proceedings of the 16th International Conference on Pattern Recognition (ICPR '2002)*. Volume 4., 2002, pp. 78 – 81.

[10] Morris, T., Blenkhorn, P., Zaidi, F.: Blink detection for real-time eye tracking. *J. Netw. Comput. Appl.*, 2002, vol. 25, no. 2, pp. 129–143.

[11] Tomasi, C., Kanade, T.: Detection and Tracking of Point Features Technical Report CMU-CS-91-132. *Image Rochester NY*, 1991, vol. 91, no. April, pp. 1–22.

[12] Viola, P.A., Jones, M.J.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: *CVPR*, IEEE Computer Society, 2001, pp. 511–518.

[13] Yan, Z., Hu, L., Chen, H., Lu, F.: Computer Vision Syndrome: A widely spreading but largely unknown epidemic among computer users. *Computers in Human Behavior*, 2008, vol. 24, no. 5, pp. 2026 – 2042.