Real-time video-based eye blink analysis for detection of low blink-rate during computer use

Matjaž Divjak, Horst Bischof

Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{divjak,bischof}@icg.tu-graz.ac.at

Abstract

This paper presents an approach for real-time remote detection of eye-blink parameters. First, a combination of boosted classifiers and Lucas-Kanade tracking is used to follow the movement of face and eyes. Then, detailed eye movement is described by normal flow. Finally, a discrete finite state machine is used to detect eye-blinks. The proposed approach is evaluated on a series of short video sequences. It shows promising eye-blink detection capabilities that could be used for software-based prevention of workplace-related disorders.

1 Introduction

In the European Community more than 40% of the today's working population use computers in their daily work. Computer use is related with static work, constrained sitting and vision problems. For example, approximately 70% of computer workers worldwide are reported to having vision problems leading towards Computer Vision Syndrome. The number of computer-related jobs is expected to increase significantly in the next decade, along with the number of workplace-related illnesses.

One of primary causes of vision problems during computer use is insufficient eye movement, caused by long periods of gazing at computer screen. Distance between the screen and the user's head usually doesn't change much and as a result the muscles involved in adaptation of the eye are not exercised for long periods of time, leading to their weakening. This is usually accompanied by decrease in eye blinking frequency, which leads to excessive dryness of the eye surface (cornea and sclera) and can be harmful to the eye. Chronic dry eyes can eventually lead to scarring of the cornea and sight loss.

With preventive measures like regular breaks, eye exercises and relaxational activities most of those disorders can be avoided. Unfortunately, the majority of population is reluctant to change their workplace habits until first signs of health issues appear. To help users become aware of the problem and assist them in prevention, we propose to use a simple monitor mounted camera (webcam) for capturing video of user at his workplace and estimating his eye blinking patterns. When potentially harmful behaviour is detected, the user can be alerted and informed about suitable actions.

Remote detection of eye-blinks from video is not as accurate as head-mounted eye trackers, but this is usually compensated by greater ease of use, non-invasiveness and much lower cost. For our purpose, the eye tracking must run in real-time, without any additional hardware (like IR illumination for example), using low quality input video, and be capable of operating under varying indoor conditions (typical office environment). Several eye tracking approaches that address those issues were already published, but few address all of the mentioned constraints and most are not suitable for precise evaluation of eye blinking movements. For example, Morris et al. [1] proposed a blink detection system based on variance map calculation and eye corner analysis. It runs in real-time on 320 × 240 images. They report good blink detection results (95 % true positives), but head movements affect the variance map computation and cause a sharp drop in performance.

Sirohey et al. [2] presented an approach for determining eye-blinks by locating eye corners, eyelids and irises and analyzing their movements. Motion information is estimated using normal flow. Head motion is modelled separately by an affine model and is used to decouple eye movements from the head movement. Authors claim their algorithm can track iris and eyelid motion correctly more than 90 % of the time, but not in real-time. In a later paper [3] they added a deterministic finite state machine (DFSM) with three states to analyze the normal flow and calculate eye blink characteristics.

Chau and Betke [4] describe a system that detects blinks in real-time using correlation with an open eye template. If large head movement occurs, the system is automatically reinitialized. For 320 × 240 images obtained from a webcam they report 95 % overall blink detection accuracy. Downside of this approach is that it distinguishes only 2 eye states, open and closed. Any movement in-between is not well defined.

Pan et al. [5] use a boosted classifier to detect the degree of eye closure. The changing of eye states is modelled by a Hidden Markov Model. The method operates in real time on 320 × 240 webcam images, detecting more than 96 % of eye-blinks. Since examples of typical eye motion are used for training the model, the method should face difficulties when non-standard eye motion occurs, such as partial blinks.

Recently, Orozco et al. [6] proposed using two appearance-based trackers: the first one tracks iris movements while the second one focuses on eyelids and blinking. Using low resolution input video and a simple appearance model, the method reportedly runs in real-time, achieving good tracking results. Authors didn't try to detect eye-blinks, however the method could be used for this purpose.

Most of the described approaches detect blinks by locating separate eye parts such as iris, eyelids and eye corners. Quality of detection is directly linked to accuracy of feature localization. When low quality webcam images are used as input, sufficient accuracy is very difficult to achieve. Additionally, real-time constraint prevents us from using sophisticated eye localization techniques.

Therefore, our approach doesn't depend on locating specific eye features. Instead, it is based on general appearance of the eye and its motion. We use a combination of appearance-based and feature-based tracking to follow the motion of face and eyes in real-time (Section 2). Eye motion parameters are estimated using normal flow (Section 3). For analysis of the flow we upgrade the DFSM-based approach presented by Sirohey et al. [2] by adding another state and including both magnitude and direction of the flow (Section 4). Performance of our approach is demonstrated on several video sequences in Section 5.

2 Detection and tracking of eyes

To detect the eyes we must first detect the face. If the camera is mounted on a computer monitor, user's face is often visible in a frontal pose. We detect it using a boosted classifier which is part of the OpenCV library [7], and was trained on a large set of frontal face images. Variations in the training set (illumination, facial expressions, glasses, body hair) provide adequate level of robustness for our purpose.

Because it is very fast, this classifier can be used for initial detection of the face as well as subsequent tracking of the face, but it cannot detect faces if they deviate significantly from the frontal pose. When this happens, we track the position of the face by a well-known Lucas-Kanade (LK) feature tracker [8]. However, tracking doesn't rely on specific facial features like eye corners, but uses a set of currently visible points with strong local contrast, detected by FAST feature detector [9]. Their position is tracked from previous to the current video frame by the LK algorithm. Displacement of feature points describes the displacement of the face. This procedure is repeated for every frame until face enters the frontal pose again, where it is detected by the face classifier. To additionally reduce the processing time, we run the classifier once for every 5 frames and track the other 4 frames by the LK tracker.

The same approach is adopted for detection and tracking of the eyes inside the facial region. Left and right eyes are detected separately by a boosted classifier. If detection is unsuccessful, the eye is tracked from its last know position using the LK tracker. We trained the eye detector ourselves using OpenCV's implementation of AdaBoost [7]. From several facial image datasets [10][11][12] with available ground-truth information for eye position we generated a positive training set of 10,000 eye images. Negative training set consisted of 13,000 facial images with eye regions removed. Figure 1 shows an overview of both training sets.

This procedure enables us to track most of the facial and eye motion performed by a computer user. One exception is when face is visible from the side, but such cases are not suitable for blink detection anyway. Very fast movements of the head are another exception, because LK tracker is not capable of following large movements and may start to lag behind the actual face location. However, the situation is corrected as soon as face enters the frontal pose again.

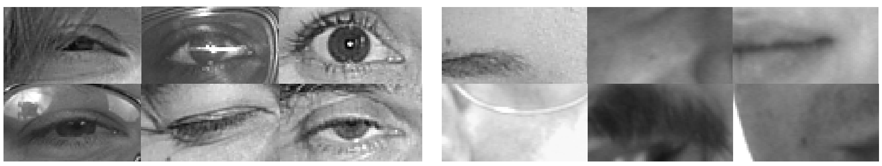

Figure 1: An overview of positive (left) and negative (right) training set for a boosted eye classifier.

3 Eye movement evaluation

Once eye regions are located we can estimate the eye movement parameters. Eye movements detected on a 2D image are composed from two independent components: global movement of the face and local movement of the eye. Since face movement is known by now, we can compensate for its effect to obtain the actual eye movement. The compensated eye motion can then be used to detect blinks.

Object motion is usually described by optical flow. Reliable estimation of optical flow is a complex procedure, not suitable for real-time operation. Following the example of Heishman and Duric [3], we calculate normal flow of the eye region in the direction of intensity gradients, where it is well defined. Then, facial movement is subtracted from the estimated normal flow. An example of such compensation is depicted in Figure 2. Twenty frames showing a person moving his head upwards while blinking at the same time were extracted from the Talking Face dataset [13] and processed by our algorithm. Left side of Figure 2 shows the estimated dominant orientation, along with X and Y components of average magnitude of the non-compensated normal flow. Facial movement clearly dominates as the direction is strongly downwards (approx. 270°) for all frames and vertical component of the flow is dominating (Figure 2, left side). However, when facial movement is compensated, the orientation reveals a dip indicating a blink, and magnitude also shows the blinking motion in X as well as Y directions (right side of Figure 2).

The size of the user's face in the input image can vary greatly, affecting the magnitude of the estimated normal flow. To prevent this, we normalize the normal flow vectors by dividing them with the size of the current face region. This makes flow from small (distant) faces comparable with the flow from large (close) faces.

Since we are only interested in closing and opening of the eyes, we focus our analysis only on flow in the direction perpendicular to a line connecting centers of left and right eye regions. The angle between this line and the horizon is calculated and flow vectors are rotated correspondingly. Corrected and normalized flow $\boldsymbol{n}$ is then used to calculate mean flow magnitude of the eye region, denoted as n_{MAG}. To detect dominant flow direction n_{DIR}, vector orientations are stored in a histogram with 36 bins, each bin representing a 10° arc. The bin with the most contributions represents the dominant orientation of the flow. If histogram has multiple peaks, their average is used.

4 Blink detection

Information from eye's normal flow enables us to detect eye blinks together with all the intermediate states such as partial blinks, squints, etc. Flow magnitude (n_{MAG}) is our main indicator, but direction of the flow (n_{DIR}) is also important. Detecting blinks by a simple thresholding of the magnitude doesn't give good results. Different speeds of blinking will give very different flow magnitudes, but the general sequence of movements will remain largely the same.

Heishman and Duric [3] use a deterministic finite state machine (DFSM) to estimate blink parameters. The three states used are: Steady (Open), Closing and Opening. This enables detection of blinks, but doesn't allow a lot of variations in eye movement, like holding eyes closed for a period of time. They also have to manually adjust several thresholds for each user. This gives the best detection results, but is inappropriate for our application since it burdens the user.

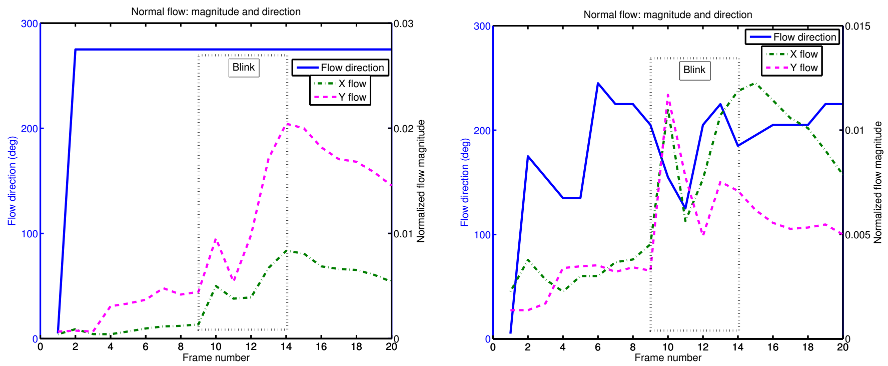

Figure 2: An example of removing head motion from the estimated normal flow of the eye region, shown on frames 1190 – 1210 of the Talking face video [13]. Left: dominant flow orientation and mean flow magnitude of initial normal flow. Right: dominant flow orientation and mean flow magnitude after subtraction of the head motion. In this sequence, the user blinks while his face is moving upwards.

Therefore, we extend this idea and use two separate states for open and closed eye. Our DSFM thus has 4 states: Open, Closing, Closed and Opening. Transitions between states are based on two parameters: mean magnitude of the normal flow n_{MAG} and dominant orientation of the flow n_{DIR}. Structure of our DFSM is showed in Figure 3. To move out of a steady state, flow magnitude must exceed a certain threshold. This threshold T is set to $T = 6 \cdot \mathit{standard\_deviation}\ (\boldsymbol{n^*})$, where $\boldsymbol{n^*}$ is normal flow, estimated only during stationary eye states. Experiments show this is suitable for well-behaved eye movements. Persons with significantly different eye dynamics need to adjust this threshold, but since it's the only important parameter it's relatively easy to do so. Left side of Figure 4 shows changing of the DFSM states and estimated T value for a simple video sequence with one slow, long blink. Selecting a threshold for flow direction is easier. Since we are only interested in upwards/downwards movement, all flow vectors with directions between 20° and 160° are classified as upward motion, and all vectors with directions between 200° and 340° are classified as downward motion. Those limits are depicted on the right side of Figure 4.

Such DFSM enables us to detect separate eye states; however, those states are often not reliable due to noise in flow data. This is particularly true for Opening and Closing states and during periods with low flow magnitude. If flow direction fluctuates, the DFSM can make incorrect transitions between states. To address those fluctuations, additional post-processing step is added that analyses the state transitions and removes spurious "blinks". This post-processing step could also be used to detect other types of eye motion besides blinking.

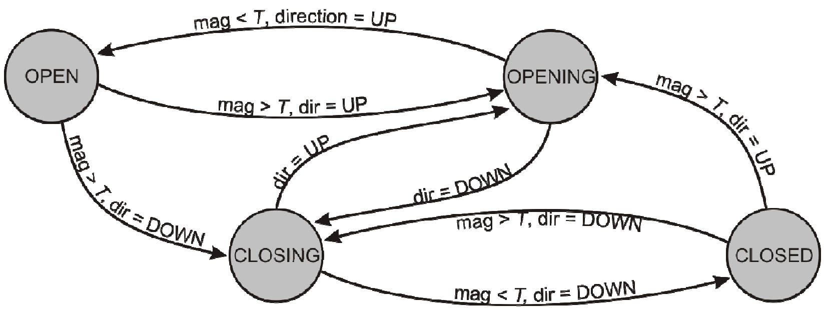

Figure 3: Structure of our discrete finite state machine for identification of eye-blinks from normal flow data. Notation: *mag* = mean flow magnitude, *dir* = dominant flow direction, T = mean flow magnitude threshold.

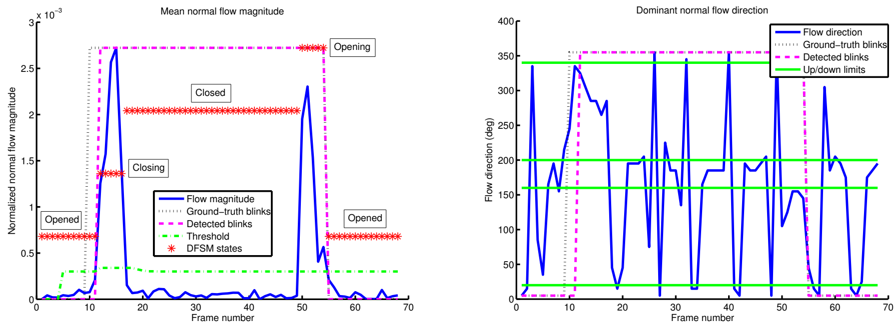

Figure 4: An example of mean flow magnitude and dominant flow direction for a video sequence showing one very slow blink. Left: mean flow magnitude n_{MAG} with marked threshold and DFSM states. Right: Dominant flow direction n_{DIR} with marked threshold for upward/downward movement.

Finally, eye-blink parameters are calculated and statistically evaluated. Currently, we observe the duration of blinks and the duration of pauses between blinks (blink frequency). If duration of blinks is significantly lengthened and blink frequency is reduced, this signals the unwanted staring at the computer screen and triggers a warning for the user. By tracking eye-blink statistics for longer time periods, users could be forewarned of hazardous working habits and assisted in timely prevention of computer-use related illnesses.

5 Experimental results

Presented approach for detection and analysis of eye-blinks was tested on several video sequences. No standard test sets exist for this purpose, so we recorded a few test videos ourselves. Additionally, the Talking Face video [13] and recently presented ZJU Eyeblink Database [5] were used. All video frames are 320 × 240 pixels in size. Ground truth data for blink detection was obtained manually by marking the starts and ends of blinks.

5.1 Face and eye tracking

Face and eye tracking accuracy is sufficiently good for our purpose, and illumination changes or presence of glasses don't affect the performance significantly. Face pose can change up to approx. 40° away from the frontal pose without much ill-effect. This is demonstrated in Figure 5.

Figure 5: Two examples of input video with detected face and eye positions. Top: natural light, glasses. Bottom: Talking Face video [13].

5.2 Blink detection

Eye-blink detection is evaluated on 15 short videos. We use our own recordings and a subset of the ZJU Eyeblink Database. Videos are very varied: they include static as well as moving faces, dynamic backgrounds, other people walking behind the user, different illumination conditions, on several videos users are wearing glasses. Blinks are detected by a two-stage analysis described in Section 4. In all tests the same value of threshold T was used. Table 1 presents the resulting eye-blink parameters.

On our computer (Intel Core 2 Duo, 2.66 GHz, 2 GB RAM) the unoptimized C++ program typically processes one 320 × 240 image in 35 ms (28.6 fps). Approximately 20 ms is spent for detection and tracking, and 15 ms is spent for normal flow calculation.

Eye-blink parameter	Estimated mean value ± std
Frames with detected blinks	87.3 % ± 8.7 %
Frames with missed blinks	12.7 % ± 8.7 %
Correctly detected blinks	94.2 % ± 14.9 %
Incorrectly detected blinks	5.8 % ± 14.9 %
Mean blink duration	239 ms ± 191 ms
Mean blink frequency	783 ms ± 672 ms

Table 1: Numerical results for eye-blink analysis on a set of test video sequences.

Results show that it is possible to obtain realistic eye-blink parameters using the proposed method. With long-term tracking of average blink duration and frequency values it is possible to detect signs of eye fatigue and to help the user in preventing it.

Unfortunately, it's very difficult to compare those results with previously published work, because there are no standard evaluation datasets and authors report very different performance metrics. If we compare our results with those from [3] we notice that despite using different video sequences the reported mean blink duration (220 ms – 390 ms) is very close to our value of 240 ms, while duration between blinks is much longer in their case (2.7 s – 4.8 s). Our improvements to the DFSM proposed in [3] are not very noticeable when detecting simple eye-blinks, but we believe the difference is more significant when analysing complex eye behaviour. Additionally, our approach runs in real-time and uses only 1 parameter that must be adjusted only if person's eye dynamics deviates a lot from default values, while approach from [3] requires 3 parameters that need to be changed across subjects and scenarios and was tested off-line.

Pan et al. [5] recently presented the ZJU Eyeblink Database that is to our knowledge the only promising attempt to provide the community with standard eye-blink evaluation data. Unfortunately, they provide only one performance metric: the mean eye-blink detection rate when using their algorithm is 95.7 % with false alarm rate < 0.1 %. Duration of blinks or blink frequency are not reported. When our program is evaluated on the whole ZJU dataset with 80 videos, the mean blink detection rate is 70 % with additional 13 % wrong detections. Those numbers are largely affected by the fact that several videos from ZJU contain eye-blinks that appear shortly after the start of the video. This confused our algorithm which needs stationary eyes at the beginning of the analysis to correctly determine the standard deviation of normal flow for threshold *T*. Therefore, several videos gave no detected blinks, which significantly skewed the results.

Conclusion

We presented an approach for video-based detection of eye-blinks that could be used to alert the computer user if potentially dangerous blinking behaviour is detected. We believe this approach could be much more effective than currently available preventive software, which is usually based on keyboard and mouse activity. The proposed two level analysis of eye's normal flow enables us to detect blinks as well as various other kinds of eye movements. To improve the performance, we plan to use GPU-based implementation of optical flow estimation instead of simple normal flow calculation. Additionally, an upgrade to the Lucas-Kanade tracker would allow for better tracking of fast, significant face movements.

Acknowledgements

The work presented in this paper is part of the "PRE-WORK" project which is supported by a Marie Curie Intra-European Fellowship within the 6th European Community Framework Programme (Contract No. 041395).

References

[1] T. Morris, P. Blenkhorn and F. Zaidi. *Blink detection for real-time eye tracking*. Journal of Network and Computer Applications, vol. 25, num. 2, pp. 129-143, April 2002.

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

[3] Ric Heishman and Zoran Duric. *Using image flow to detect eye blinks in color videos*. Proceedings of the Eighth IEEE Workshop on Applications of Computer Vision, pp. 52, 2007.

[4] Michael Chau and Margrit Betke. *Real Time Eye Tracking and Blink Detection with USB Cameras*. Boston University Computer Science Technical Report No. 2005-12, May 2005.

[5] Gang Pan, Lin Sun, Zhaohui Wu and Shihong Lao. *Eyeblink-based anti-spoofing in face recognition from a generic webcamera*. The 11th IEEE International Conference on Computer Vision (ICCV'07), Rio de Janeiro, Brazil, October 2007.

[6] Javier Orozco, F. Xavier Roca and Jordi Gonzalez. *Real-time gaze tracking with appearance-based models*. Machine Vision and Applications, 2008.

[7] Rainer Lienhart and Jochen Maydt. *An Extended Set of Haar-like Features for Rapid Object Detection*. IEEE ICIP 2002, vol. 1, pp. 900-903, Septembre 2002.

[8] Jean-Yves Bouguet. *Pyramidal Implementation of the Lucas Kanade Feature Tracker*. Intel Corporation, 2002.

[9] Edward Rosten and Tom Drummond. *Machine learning for high-speed corner detection*. European Conf. on Computer Vision, vol. 1, pp. 430-443, May 2006.

[10] *The BioID Face Database*, HumanScan AG, Switzerland, http://www.bioid.com/downloads/facedb/index.php.

[11] *The Color FERET Database*, Version 2, National Institute of Standards and Technology, http://face.nist.gov/colorferet/.

[12] N. A. Fox, B. A. O'Mullane and R. B. Reilly. *The Realistic Multi-modal VALID database and Visual Speaker Identification Comparison Experiments*. Proc. of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA-2005), New York, July 20-22, 2005.

[13] *Talking Face Video*. Face&Gesture Recognition Working group, IST-2000-26434, http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_ face.html.