Eye Blink Detection using Variance of Motion Vectors

Tomas Drutarovsky, Andrej Fogelton

Vision and Graphics Group
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava, Slovakia
{xdrutarovsky,andrej.fogelton}@stuba.sk

Abstract. A new eye blink detection algorithm is proposed. It is based on analyzing the variance of the vertical component of motion vectors in the eye region. Face and eyes are detected with Viola – Jones type algorithm. Next, a grid of points is placed over the eye regions and tracked with a KLT tracker. Eye regions are divided into 3×3 cells. For each cell an average motion vector is estimated from motion vectors of the individual tracked points. Simple state machines are setup to analyse these variances for each eye. The solution is this way more robust and with a lower false positive rate compared to other methods based on tracking. We achieve the best results on the Talking face dataset (mean accuracy 99%) and state-of-the-art results on the ZJU dataset.

Keywords: statistical variance, local motion vectors, outlier detection, KLT tracker

1 Introduction

Eye blink detection has different uses e.g.: driver fatigue detection [1], computer user monitoring system to prevent dry eye syndrome [5], helping disabled people to interact with computer [9] or face liveness detection [15].

Eye blink is defined as a short unintended move of eyelids [13]. Eyelids are closed for at least 50 ms during the fastest eye blink. Eye blink in general lasts from 100 ms to 300 ms. These numbers indicate that a standard camera with 25-30 frames per second (fps) is sufficient and also it allows a real-time performance. There are two possible ways to detect eye blinks. Sequential methods based on a motion tracking in the eye region [5] or computing difference between frames (pixels values [11], descriptors [4], etc.). Appearance based methods estimate the state of the eye (open, closed [12] or the eye closure [7]) for individual frames, based on which individual eye blinks are detected.

In our experience appearance based methods have often difficulties with different conditions as a thick frame of glasses, a strong eye brow or low percentage of eye openness (race dependent) etc. We created a sequential eye blink detection method, which uses Viola – Jones type algorithm [17] to detect face and eye regions. Points are placed over the eye regions and tracked by a KLT tracker [16].

After outlier detection, 9 local motion vectors are estimated within the eye region. The variance of the vertical components of these motion vectors is used as the input for the state machine. If at least one of the state machines (left and right eye) changes to the eye blink state, the method will evaluate it as an eye blink is being detected. The best results are achieved on the *Talking face* dataset[1], achieving 99% of mean accuracy and comparable results on the ZJU dataset[2].

The rest of the paper is structured as follows; Section 2 introduces related work. Section 3 describes the introduced method. Section 4 presents the evaluation on available datasets, introduces our new challenging dataset and discusses the results.

2 Related Work

Majority of methods are modularized as follows; Viola – Jones type algorithm detects face and eyes, next eyeblink detection is performed e.g. [7, 3]. Complementary face, eye region tracking is often performed [12, 2].

Intensity Vertical Projection (IVP) [4] is the total pixel intensity in a frame row. The method uses the fact that an iris has a lower intensity vertical projection values compared to other regions. IVP function has two local minimums considering one eye. Blink occurs when the IVP curve changes.

The method in [1] measures ocular parameters by fitting two ellipses on eye pupils using the modification of the algebraic distance algorithm for conics approximation [6]. The degree of an eye opening is characterized by the pupils shape. To eliminate the inaccuracy of the algorithm for fitting ellipses, the state machine is defined to detect eye blinks. In [7] the percentage of closure is calculated from the ratio between the iris height in the frame and the nominal value assigned during a ten-second calibration. This detector reaches 90% recall and low false positive rate on their own database consisting of 25 hours of driving records.

[12] introduces two features; $F1$ as the cumulative difference of black pixels in the eye region estimated from the binary image from the successive frames. The authors observed that the number of black pixels in a closed eye image is higher compared to an open eye image. There is a problem with the estimation when the tracking subject is moving closer to the camera and back. To avoid it, the method uses the cumulative difference counting with an adaptive threshold. This threshold changes due to the accumulated difference over time. The second feature $F2$ represents a ratio of eye height to eye width. To calculate $F2$, a binarized eye region is processed through an erosion and dilation filters. The eye state (open, closed) is estimated by a maximal vertical projection of black pixels. An open eye has greater value of this ratio, because it has higher maximal projection value (Figure 1). For the correct eye openness estimation,

[1] `http://www-prima.inrialpes.fr/FGnet/data/01-TalkingFace/talking_face.html` [accessed: 22.7.2014]

[2] `http://www.cs.zju.edu.cn/~gpan/database/db_blink.html` [accessed: 27.8.2014]

the authors use the features $F1$ and $F2$ as the input values to the SVM. Three SVM classifiers are used for the three different rotation angles to determine the eye state of the subject.

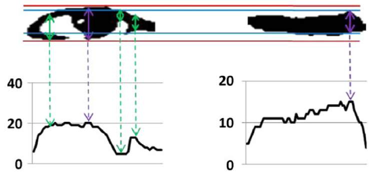

Fig. 1. An open eye has greater value of the ratio of eye height to eye width, because it has higher the maximal projection value. Purple arrow signifies the highest intensity projection value [12].

In [14] eyelids are detected as follows: first, an image is divided into several vertical sections. For each section, candidates for upper and lower eyelids are defined as the maximal and minimal differential values of the gray level distribution. These candidates are grouped in five sections, two of them are chosen to represent the upper and lower eyelid. All five sections are then used to calculate the *eye gap* – an average of distances between eyelid candidates. The eye gap is the degree of eye openness. Over time, it represents a blink waveform. The eye gap decreases rapidly when eye blinks. After the eye gap reaches the minimum value (eye is considered as closed), it increases gradually.

In [3], the neural network-based detector is used for the precise eye pupil localization. The head rotation angle is calculated using the vertical positions of both pupils. Pupils are analyzed using the horizontal symmetry calculation to determine whether the eyes are *Open* or *Closed*. Pupil region is divided in two halves using the axial symmetry around the line crossing centers of both pupils. Created halves represent the upper and lower eyelid regions. These halves are fold over. If the eye is open, then the folded fragment preserves its circular shape symmetry, unlike the closed eye. Therefore, difference between the upper and lower half is estimated. The algorithm is tested on the ZJU dataset and it achieves 94.8% of mean accuracy.

In [8, 10], eyes are detected using the correlation coefficient over time. The open eye template is learned to estimate the eye closure and detect eye blinks. Reinitialization is triggered by the correlation coeficient falling under the defined threshold. According to the changing correlation score of the eye and its open eye template, an eye blink waveform is established. The correlation score is binarized: open and closed eye.

[5] computes the optical flow of face region proposed in [18] and compensate the face movement. Optical flows for feature points in the eye region are normalized and the rotation of flow vectors is computed to estimate the main flow direction. Eye blinks are detected using an adaptive threshold on the main flow size.

3 Eye Blink Estimation by Statistical Variance

In the proposed method we first run Viola – Jones type face and eye detection algorithm (OpenCV[3] implementation, frontal face and eye pair) over the image. Then the eye region is enlarged in height (1.5×) to cover larger area and to compensate the inaccuracy of the eye region detection. If the eyes are not detected, given frame is skipped. The Eye region is divided into halves to separate individual eye regions. Regular grids of points are placed over. Approximately 225 points are distributed each 1/15 of width and height for each region. Points are tracked by a KLT tracker. Workflow of the algorithm is visualized in Figure 2.

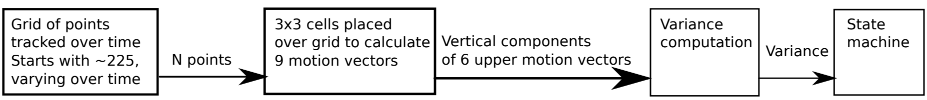

Fig. 2. Workflow of the tracking and eye blink detection of the proposed method

3.1 Eye Center and Bounding Box Estimation

Points predicted by a KLT tracker in following frame are first verified using the optical flow control. If the optical flow is not found, such a point is removed from the tracking. We use the KLT tracker implementation with image pyramid and thus it can sometimes predict highly unprobable locations (major shifts relocation across image). Therefore we remove points which change their location by more than a half of the image height from future iterations.

The remaining outliers are filtered based on their euclidean distance from the eye center. The eye center is calculated as the average point from all feature points for each eye. Histogram of individual point distances from the center is created. We experimentaly evaluated on our dataset an *outlier_treshold* as the beginning of at least 3 bins in a row after the global histogram maximum (Figure 3).

Outliers are removed from further analysis only, their tracking continues. Based on our observations these outliers can return to their previous locations in next frames. Tracked person can move towards and backwards from the camera, thus the eye regions are reestimated each frame. Square bounding box is placed in the eye center with side length of $1.6 \times$ *outlier_threshold*.

3.2 Analysis of Motion Vectors

The eye bounding box is divided into 3×3 cells (Figure 4). During eye blink we observe (Figure 6) a significant vertical move in the middle cells (numbers

[3] `opencv.org`

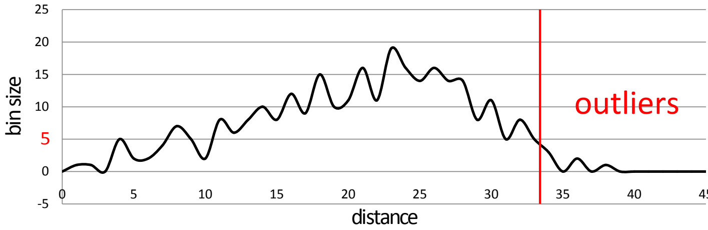

Fig. 3. Ilustrative histogram of feature point distances from the eye center. Threshold is detected after the global maximum, where at least 3 bins in a row have feature points count under threshold (5).

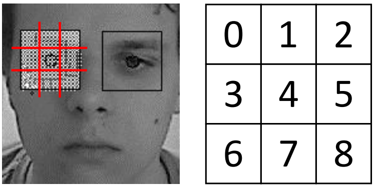

Fig. 4. Eye bounding box is divided into 3×3 cells.

3, 4 and 5), but only minor vertical move in the upper cells (0, 1, 2). We use vertical component of motion vectors of these 6 first cells 6. Vertical component is sufficient, because there is a strong predisposition, that the person's face does not rotate significantly.

Statistical variance of these 6 cells is calculated to evaluate the diversity of moves across them. If the variance is higher than the *variance threshold* T, it indicates an eye lid movement. Variance is invariant to position changes of the person's face and therefore no head movement compensation is necessary. The *variance threshold* is evaluated empirically on our dataset as $T = dist \times \frac{1}{fps} \times const$, where:

- *Eye distance (dist)*: Distance between the eye centers is directly proportional to the subject distance from the camera. The eye size affects the size of motion vectors during eye blink.
- *Frames per second (fps)*: Frame rate of the input video sequence also influences the size of motion vectors. Higher frame rate means lower variance of motion vectors.
- *Constant*: Constant value is set based on the tests on our dataset as 0.00068.

Motion vectors represent local move in the eye region, which is a sign of an eye blink or other moves such as (facial mimics, eyebrow or iris). To eliminate detection of such moves a simple state machine is set up (Figure 5). We consider eye blink as the conjugation of two moves – down and up move. After move down detection, the state changes from *State 0* (the initial one) into *State 1*. When the state machine is in *State 1* for about 100 ms (3 frames while 30 fps) and move up is detected, the state changes into *State 2* – blink occurred. If there is no move for more than 150 ms (5 frames while 30 fps), the state machine changes from *State 1* back to *State 0*. Different sizes of

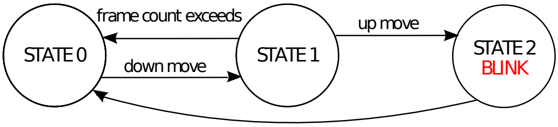

Fig. 5. The state machine for blink detection. After down move detection, the state changes from State 0 into State 1. While in State 1 and move up detection, the state changes into State 2 – blink occurred. State 1 changes back to State 0, if there is no move for given amount of time.

vertical components of motion vectors across cells are visualized in Figure 6. Sizes are normalized based on Cell 1. Based on our observation we define move down as `(cell4 > 0 && (cell4-cell1) > 0 && variance > T )` and move up as `(cell4 < 0 && (cell4-cell1) < 0 && variance > T )`. Algorithm 1 presents the state machine pseudo code for eye blink detection. There are two state machine established, one for each eye. Eye blink is detected if at least one of the machines detects it.

I) Down head move

80%	100%	63%
66%	86%	66%

II) Down blink move

49%	100%	-8%
270%	301%	71%

(a) During down head move (I) all motion vectors have similar size. During eye lid move down (II), cell 4 reflects the most significant move.

III) Up head move

-133%	-100%	-80%
-98%	-96%	-67%

IV) Up blink move

-161%	-100%	-221%
-83%	-940%	-683%

(b) During up head move (III) all motion vectors have similar size. During eye lid move up (IV), cell 4 reflects the most significant move (different direction).

Fig. 6. Visualization of vertical move in the first 6 cells during head and eye blink moves. Cell number 1 is used as the reference one for normalization.

3.3 Reinitialization

Even distribution of tracked feature points is negatively effected (Figure 7) by move in general. Even distribution is important to acquire representative data of local motion vectors. We reinitialize our algorithm with eye detection in case of following events:

– large number of lost feature points: more than half of the tracked points is lost between two frames or the remaining number of the points is less than 20.

Algorithm 1 The state machine to detect eye blink based on variance of vertical component of motion vectors in upper 6 cells of an eye bounding box.

INPUT: current and previous moves vector, current state of the state machine, distance between eye centers, FPS count
OUTPUT: eye blink status – boolean value

```
procedure CELL_METHOD_STATUS(current, previous, state, distance, fps)
    for i ← 0, 5 do                      ▷ Computer vertical component of motion vectors
        change(i) ← current(i) − previous(i)
    end for
    T ← distance ∗ const/fps                                        ▷ variance threshold
    upper ← change(1)
    lower ← change(4)

    if state = 0 then
        if move in positive direction and GET_VARIANCE(change) > T then
            state ← 1                                        ▷ down eye-lid move occurred
            time = startMeasure()
        end if
    else if state = 1 and time > 100 ms and time < 150 ms then
        if move in negative direction and GET_VARIANCE(change) > T then
            state ← 0                                          ▷ up eye-lid move occurred
            return true                                            ▷ eye blink occurred
        end if
    end if
    return false                                               ▷ eye blink did not occur
end procedure
```

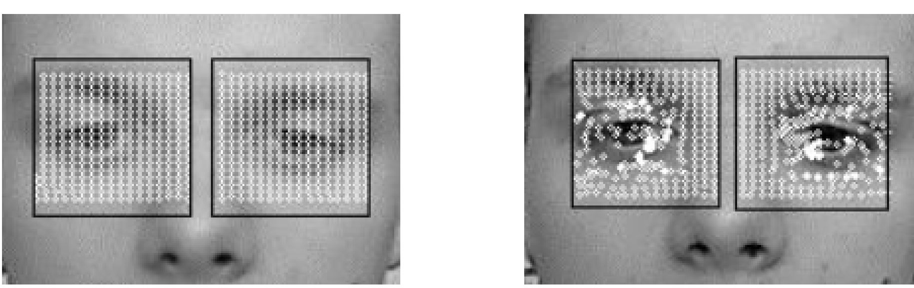

Fig. 7. Left figure shows initial grid of feature points tracked by the KLT tracker. Right figure represents their positions after 3 blinks. Many points are snapped to corners and edges and it disrupts even distribution. Reinitialization is necessary, because of even distribution of feature points.

- after blink occurred.
- over time, after constant number of frames (in our implementation every 200 frames).

Reinitialization restores all parameters to its initial values. However, it is necessary to preserve the current *State* of the state machine and blink frames count, which preserves the information of how many frames in a row with no movement was observed during the *State 1*. This way the reinitialization will not abort the eye blink detection process.

4 Evaluation

We introduce a new dataset called *eyeblink8*, which consists of 8 videos (Figure 9) with 4 individuals (1 wearing glasses). Videos are recorded under different conditions with faces mostly oriented directly to the camera. It contains many natural face movements, facial mimics and other intensive non-blink moves. The dataset contains over 82600 frames (640×480) and 353 blinks. All videos are recorded using Logitech C905 camera with 30 fps acquisition.

Annotation to individual videos consists of two files; the first file contains frame numbers with acquisition time and the second file are manually annotated states of the eye. We recognize 3 states: open, half and close. When the blink starts, *half* tags are assigned to individual frames until is fully closed. Fully closed eyes are tagged with *close* and openning eye is again tagged with *half* until is fully open. Also not fully closed eye blinks can be annotated this way (consisting only from "halfs"). If only one eye blinks (or is visible) tag *Left/Right* is added to the eye state. Sample annotation is in Figure 8. Eye blink is considered as detected if there is any intersection interval between the detected blink and anotation. For now, we do not use the information about the start of the eye blink from the state machine, we create an artificial interval around the frame with the detected eye blink with 3 frames on each side (7 frame interval with the detected eye blink in the middle). The intersection interval can be counted just once as *True Positive.*

frame number	acquisition time	frame number	state	frame number	state
2036	72.5077	8433	half	3643	halfRight
2037	72.5392	8434	half	3644	halfRight
2038	72.5712	8435	close	3645	halfRight
2039	72.6032	8436	close	3646	closeRight
2040	72.6356	8437	half	3647	closeRight
2041	72.6672	8438	half	3648	halfRight
2042	72.6994	8439	half	3649	halfRight

Fig. 8. Sample annotation for dataset *eyeblink8*

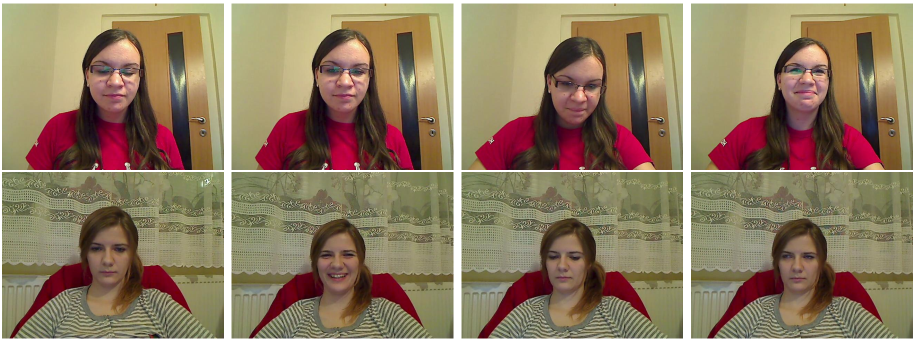

Fig. 9. Sample snapshots from our dataset the *eyeblink8*

We evaluated our method also on 2 publicly available datasets (Talking, ZJU). *Talking Face Video (Talking)* (Figure 10) contains 5000 frames (720×576) with 61 eye blinks. A tested subject is a man taking conversation during the record. His face is mostly oriented directly to the camera and slightly turned aside, what causes minor problems with precise eye detection. We created new annotation compatible with our evaluation framework described above. Results are compared with existing methods in Table 1. We failed to detect only two blinks, which happened during the downward sight, therefore the size of vertical component of motion vectors is not significant enough.

Fig. 10. Sample snapshots from the *Talking face* dataset

The ZJU dataset (Figure 11) consists of 80 short videos (10876 frames) of 20 individuals with and without glasses (insignificant small frame) captured with 30 fps and size of 320×240. The ZJU contains together 255 eye blinks collected indoor, some of them are voluntary longer eye blinks. It is interesting that the annotation to this dataset from [3] contains even 272 eye blinks. We also manually annotated this dataset and based on our eyeblink definition there is 264 eye blinks. When the eyelid moves down and up it is not always an eye blink. This dataset contains also one frame blinks and sometimes people blink twice very fast, we consider these double blinks as two. It is possible that the original annotator did consider these events differently. In Table 1 the number next to the ZJU is the number of ground true eye blinks.

Fig. 11. Sample snapshots from the *ZJU* dataset

Comparison on available datasets is presented in Table 1. Methods we compare with do not mention how they calculated *false positive rate* and *Mean accuracy*. We assume that the number of images with open eyes is used as *Negatives* (N). In our opinion this is not accurate, because an average eye blink can be observed on 7 frames in a row in average. We divided the number of frames with open eyes in datasets by the average blink duration and this is used as true negative (number of non eye blinks). From our annotation of the ZJU dataset can be read, that 2482 frames capture some part of eye blinks. This is used to calculate more precise *False Positive* (FP) rate and *Mean accuracy* (MA). We use the following equations: $Precision = \frac{TP}{TP+FP}$, $Recall(TPrate) = \frac{TP}{TP+FN}$, $FPrate = \frac{FP}{N}$, $MA = \frac{TP+TN}{P+N}$.

Table 1. Comparison of our method on the eyeblink8, the ZJU and the Talking dataset. The number next to the ZJU represents the number of ground true eye blinks. There are two results in FP rate and Mean accuracy, because as true negative we consider a non blink action and not an image with open eyes (the result in brackets), which we assume is used to calculate the results in papers we compare to.

	Dataset	Precision	Recall	FP rate	Mean accuracy
Divjak & Bischof [5]	Talking	-	95%	19%	88%
Divjak & Bischof[4]	Talking	-	92%	6%	93%
Lee et al. [12]	Talking	83.3%	91.2%	-	-
Our method	Talking	**92.2%**	**96.7%**	**0.7% (0.1%)**	**99%(99.8%)**
Divjak & Bischof [5]	ZJU 255	-	**95%**	2%	97%
Lee et al. [12]	ZJU 255	**94.4%**	91.7%	-	-
Danisman et al. [3]	ZJU 272	90.7%	71.4%	1%	94.8%
Our method	ZJU 264	91%	73.1%	1.58%(**0.17%**)	93.45%(**99.8%**)
Our method	eyeblink8	79%	85.27%	0.7%(0.1%)	99.5% (99.9%)

4.1 Discussion

Our method achieves the best results in all metrics on Talking face dataset, but lower *Recall* on the ZJU dataset. We are unable to detect 71 eye blinks on the

[4] `http://www.icg.tugraz.at/Members/divjak/prework/PreWork-Results` [last access: 27.6.2014]

ZJU, one third is caused by inaccuracy of Viola – Jones type algorithm. Around 20 blinks are not complete, because the video starts with a person with closed eyes. Other failures occur mostly because of very fast eye blinks so the state machine is not registering it. If the video is not recorded properly, some eye blinks are only seen in one or two frames. The fastest eye blinks have closed eye for at least 50 ms [13], adding move down and up frames, the fastest eye blink should be at least on 5 frames (while 30fps).

Divjak & Bischof [5] have quite high false positive rate on the Talking face video, unfortunately they do not bring details on their head move compensation algorithm. They have very low false positive rate in the ZJU dataset, where people are calm and not using face mimics and movements as in the Talking face.

Our method is invariant to shifting and thus we do not face the problem of head move compensation. It has to be stated that very fast head nodding is still detected as false positive.

Lee et al. [12] achieves the best precision on the ZJU dataset but by 10% lower on the Talking. Their method is an apearance based and we assume that is capable of detecting the 20 incomplete eye blinks which are in the beginning of the videos. We notice that low precision in Talking dataset could be caused by significant eye brows of the person, which are closer to the eye as in the ZJU dataset. The Talking face is European person and the ZJU dataset consists of Asian people mostly whose eye brows are in average further from the eye. Their method was trained on their own dataset consisting of Asian type people facing to the camera. In *Talking face* the person often looks down, which also decreases the precision.

Our method is implemented in C++ using OpenCV achieving real-time performance on Intel Core i5 (4 cores) 3.1Ghz with 20% of CPU usage in average.

5 Conclusions

There is an increasing attentation on eye blink detection algorithms for different purposes as driver fatigue detection, face liveness detection etc. We present simple method based on feature point tracking and local motion vector estimation. Standard camera with 25 – 30 fps is sufficient. By using the statistical variance of the vertical component of these motion vectors as the input for the state machine we created a robust method and achieve invariance to common head moves and facial mimics. Most of the thresholds are adaptive based on the person distance from the camera and fps.

We introduce our new challenging dataset eyeblink8 with available annotations. We achieve the best results on the Talking face dataset, 99% of mean accuracy. We propose a different way to compute false positives and mean accuracy based on non eye blink and the number of images containing an open eye. We achieve state-of-the-art results on ZJU dataset.

Acknowledgment. This work was partially supported by VEGA 1/0625/14.

References

1. Bergasa, L., Nuevo, J., Sotelo, M., Barea, R., Lopez, M.: Real-time system for monitoring driver vigilance. IEEE Transactions on Intelligent Transportation Systems 7(1), 63–77 (2006)
2. Brandt, T., Stemmer, R., Rakotonirainy, A.: Affordable visual driver monitoring system for fatigue and monotony. In: IEEE International Conference on Systems, Man and Cybernetics 2004. vol. 7, pp. 6451–6456 (2004)
3. Danisman, T., Bilasco, I., Djeraba, C., Ihaddadene, N.: Drowsy driver detection system using eye blink patterns. In: Machine and Web Intelligence (ICMWI), 2010 International Conference on. pp. 230–233 (2010)
4. Dinh, H., Jovanov, E., Adhami, R.: Eye blink detection using intensity vertical projection. In: International Multi-Conference on Engineering and Technological Innovation: IMETI 2012 (2012)
5. Divjak, M., Bischof, H.: Eye blink based fatigue detection for prevention of computer vision syndrome. In: IAPR Conference on Machine Vision Applications. pp. 350–353 (2009)
6. Fitzgibbon, A.W., Fisher, R.B.: A buyer's guide to conic fitting. In: Proceedings of the 6th British conference on Machine vision (Vol. 2). pp. 513–522 (1995)
7. Garcia, I., Bronte, S., Bergasa, L., Almazan, J., Yebes, J.: Vision-based drowsiness detector for real driving conditions. In: Intelligent Vehicles Symposium (IV), 2012 IEEE. pp. 618–623 (2012)
8. 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. vol. 1, pp. 1010–1017 (2001)
9. Grauman, K., Betke, M., Lombardi, J., Gips, J., Bradski, G.: Communication via eye blinks and eyebrow raises: video-based human-computer interfaces. Universal Access in the Information Society 2(4), 359–373 (2003)
10. Królak, A., Strumiłło, P.: Eye-blink detection system for human computer interaction. Universal Access in the Information Society 11(4), 409–419 (2012)
11. Kurylyak, Y., Lamonaca, F., Mirabelli, G.: Detection of the eye blinks for human's fatigue monitoring. In: IEEE International Symposium on Medical Measurements and Applications Proceedings. pp. 1–4 (2012)
12. Lee, W.O., Lee, E.C., Park, K.R.: Blink detection robust to various facial poses. Journal of Neuroscience Methods (2010)
13. Stern, J.A., Walrath, L.C., Goldstein, R.: The endogenous eyeblink. Psychophysiology 21(1), 22–33 (1984)
14. Suzuki, M., Yamamoto, N., Yamamoto, O., Nakano, T., Yamamoto, S.: Measurement of driver's consciousness by image processing - a method for presuming driver's drowsiness by eye-blinks coping with individual differences. In: IEEE ICSMC. vol. 4, pp. 2891–2896 (2006)
15. Szwoch, M., Pieniek, P.: Eye blink based detection of liveness in biometric authentication systems using conditional random fields. In: Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, pp. 669–676 (2012)
16. Tomasi, C., Kanade, T.: Detection and tracking of point features. Tech. rep., Computer Science Department, Carnegie Mellon University (April 1991)
17. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision 57(2), 137–154 (May 2004)
18. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L1 optical flow. In: Proceedings of the 29th DAGM conference on Pattern recognition. pp. 214–223 (2007)