Eye Blink Detection Using Variance of Motion Vectors

Tomas Drutarovsky and Andrej Fogelton(✉)

Vision and Graphics Group, Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava, 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 motions in the eye region. The face and eyes are detected with a Viola–Jones type algorithm. Next, a flock of KLT trackers is placed over the eye region. For each eye, region is divided into 3×3 cells. For each cell an average "cell" motion is calculated. Simple state machines analyse the variances for each eye. The proposed method has lower false positive rate compared to other methods based on tracking. We introduce a new challenging dataset *Eyeblink8*. Our method achieves the best reported mean accuracy 99 % on the Talking dataset and state-of-the-art results on the ZJU dataset.

Keywords: Eye blink detection · Statistical variance · Motion vectors · Outlier detection · Global movement compensation

1 Introduction

Eye blink detection has different uses e.g. driver fatigue detection [1], a user monitoring for dry eye syndrome prevention [6], helping disabled people to interact with a computer [9] or face liveness detection [17].

Eye blink is defined as rapid closing and reopening of the eyelid. We focus on detection of endegenous eye blinks. Partial closed eye is called an incomplete blink. Eye blink in general lasts from 150 to 300ms [15]. Thus a standard camera with 25–30 frames per second (fps) is sufficient for eye blink monitoring. A real-time performance is desired. Published methods can be categorized into two groups: appearance and sequential based. Sequential methods are based on a motion tracking in the eye region [6] 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 [8]) for individual frames.

In our experience, appearance based methods have often difficulties with different conditions as the presence of glasses with a thick frame, a strong eyebrow or an eye openness (race dependent) etc. Tracking based [6] detector with no appearance knowledge achieves one of the best true positive rates on publicly available datasets, on the other side the false positive rate is higher due to face mimics and head movements.

L. Agapito et al. (Eds.): ECCV 2014 Workshops, Part III, LNCS 8927, pp. 436–448, 2015.
DOI: 10.1007/978-3-319-16199-0_31

We focused on tracking based method with the aim to decrease the false positive rate. Instead of the head movement compensation (which provide insufficient accuracy), we obtain the movement information from statistical variance of motion vectors.

The proposed eye blink detection algorithm consists of four steps. In the first step a flock of KLT trackers [18] is placed over the eye region. The second step uses the local motion vectors to estimate "cell" motions. The third one calculates the variance of the vertical components of these motion vectors. The variance is the input for the state machine which is the last step to detect blinks. There is a separate procedure for each eye. Blink state in one of the state machines indicates an eye blink. The method is reinitialised by a Viola–Jones type algorithm [19] to detect face and eye regions after each detected blink. This can be considered as an interplay between the flocks of trackers in the eye region and the eye detector.

The rest of the paper is structured as follows; Section 2 reviews related work. Section 3 describes the proposed method. Section 4 presents the evaluation on the publicly available datasets and discusses the results.

2 Related Work

The majority of methods are initialized with a Viola–Jones type algorithm to detect the face and eyes e.g. [3,8]. Based on circumstances, the detector is not able to detect non frontal faces/eyes, which is often compensated with region tracking [2,12]. Different approaches are used to detect eye blinks.

2.1 Appearance Based

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 minima representing an open eye. Blink is detected based on the changes of the IVP curve.

The method in [1] measures ocular parameters by fitting two ellipses to eye pupils using the modification of the algebraic distance algorithm for conics approximation [7]. The degree of an eye openness is characterized by the pupil shape. To eliminate the inaccuracy of the algorithm for fitting ellipses, a state machine is defined to detect eye blinks. In [8] 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 the author's own database consisting of 25 hours of driving records.

The best precision on ZJU dataset is achieved in [12]. The authors introduce two features using binarized image; *F1* as the cumulative difference of black pixels in a detected eye region, estimated from the binary image from consecutive frames. The authors observed that the number of black pixels in a closed eye image is higher compared to an open eye image. The number of black pixels is, however, also influenced by the distance from the camera. To avoid it, the method uses an adaptive threshold based on a cumulative difference. 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. To estimate eye openness precisely, the authors use the features *F1* and *F2* as input values to a SVM. Three SVM classifiers are used for three different rotation angles to determine the eye state of the subject.

In [16] eyelids are detected as follows: first, an image is divided into several vertical sections. In 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 closed), it increases gradually.

In [3], a neural network-based detector is used for precise eye pupil localization. The head rotation angle is calculated using vertical positions of both pupils. The region of interest is analyzed using horizontal symmetry to determine whether the eye is open or closed. The 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 horizontally flipped. If the eye is open, then the horizontally flipped fragment preserves symmetry, unlike the closed eye, because of eyelashes. Therefore the difference between the upper and lower half is used as the discriminative feature to detect closed eye. The algorithm is tested on the ZJU[1] dataset and it achieves 94.8% of mean accuracy.

In [9,10], eyes are detected using the correlation coefficient over time. Open eye template is learned to estimate the eye closure and detect eye blinks. Reinitialization is triggered by the correlation coeficient falling under a defined threshold. According to the changing correlation between the eye and its open eye template, an eye blink waveform is established. The correlation score is binarized: open and closed eye.

One of the most successful methods is Weighted Gradient Descriptor [13], which calculates partial derivatives per each pixel in the eye region not only in vertical direction but also over time. Two vectors are calculated and the distance in between them over time is used as a feature function. Using zero crossing on this function eye blinks are detected. Evaluation on ZJU dataset achieves just one false positive and very high Recall 98.8%. We do not include these results in our comparison, because the authors tuned the parameters per dataset (their own and ZJU) and reported the best achieved results.

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

2.2 Sequential Based

Eyelid motion is used to detect blinks in [5]. Eye detection runs every 5^{th} frame. Features are detected using FAST [14] and tracked with a KLT tracker. Features are classified based on their location; face, left and right eye. Eye and face regions are tracked based on the features. The authors calculate normal flow of the regions in the direction of intensity gradients. Eyelid motion also includes head movements, thus compensation based on the already extracted head movement is provided. Dominant orientations of the local motion vectors for the individual classes are extracted from a histogram of orientations, due to which partial invariance to eye orientation is achieved. To filter the eyelid motion, only the flow in the direction perpendicular to the line segment between the eyes is considered. The angle between this line and the horizon is calculated and flow vectors are transformed correspondingly. Corrected and normalized flow is used to calculate a mean flow magnitude of the eye regions. The dominant flow direction is recognized based on the individual orientation of local motion vectors (optical flow) in a histogram with 36 bins, each bin represents 10 degrees. Normal flow orientation and magnitude is used as the input parameter for state machines. Evaluation of the method on publicly available datasets can be found in [6].

3 Eye Blink Detection Using Variance of Motion Vectors

The proposed eye blink detection method assumes that the face and eye regions are localised. In the experiments reported in Section 4, the OpenCV implementation of a Viola–Jones type algorithm (we use cascade files: the frontal face and eye pair) is used. The detected eye region (both eyes) is enlarged in height (1.5×) to approximately half of the interocular distance to cover a larger area and thus to compensate the inaccuracy of the eye region detection. If the eyes are not detected, the frame is skipped. The region is divided into halves to separate individual eye regions. The method runs separatly for each eye. A flock of KLT trackers is placed over a regular grid (Fig. 1) spaced with 1/15 of the region dimensions (all together around 225 trackers, that count depends on the region size). Next, local motion vectors are extracted and averaged based on their locations. An average variance of vertical components of the 6 upper motion vectors is the input to a state machine, which detects an eye blink.

3.1 Eye Regions

The tracked person can move towards and backwards from the camera thus the eyes change their size and locations. The initial eye regions are obtained and a flock of trackers are placed regularly over them. Until the next reinitialization, the eye regions are re-estimated using the trackers for each frame.

Some of the trackers are lost over time. The trackers which fail to establish their new locations are omitted from further processing. The OpenCV KLT

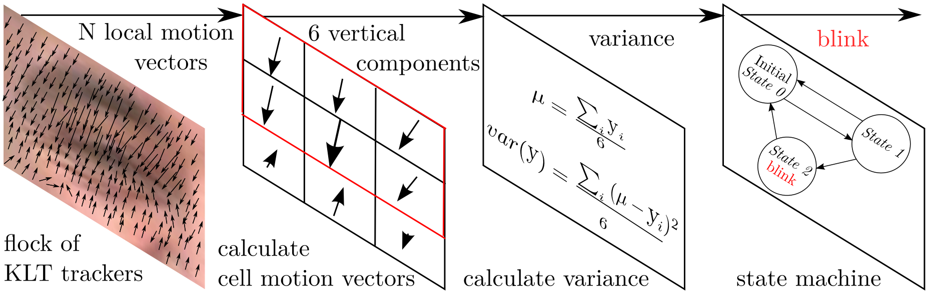

Fig. 1. Workflow of the proposed eye blink detection method

tracker implementation with an image pyramid enabled is used, from which highly unprobable new locations (major shifts across image) can occur. Therefore we omit trackers that change their location by more than a half of the image height.

KLT trackers are attracted by strong edges and corners from which some trackers diverge from the eye. We do not want to use those trackers in order to calculate motion vectors but we still keep them because there is a high probability they will return closer to the eye in future. We filter these trackers based on the estimated eye region.

The eye region is defined as a square, placed in the eye center, which is calculated as the average location of trackers in the frame for each eye. A histogram of euclidean distances of individual trackers from the eye center is created. We experimentaly evaluated a distance threshold T_d that is the beginning of the interval of 3 bins in a row with count below the threshold T_b after detected the global maximum (Fig. 2). The global maximum condition is just a precaution because the flock of trackers behavior can also cause low counts to appear at the

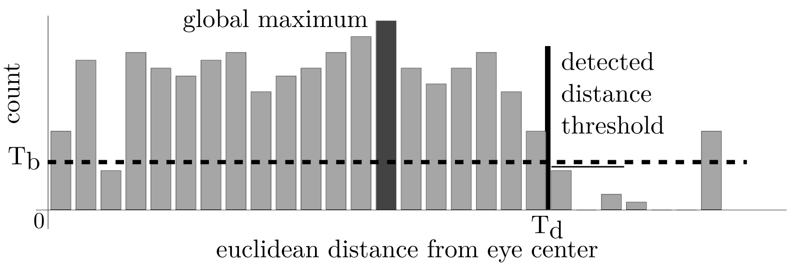

Fig. 2. Histogram of tracker euclidean distances from the eye center. The histogram is used to filter out the trackers, which are probably not in the eye region anymore. In our case, the histogram bins represent pixel distances. The distance threshold T_d is defined as the beginning of the interval of the 3 bins with count below the threshold T_b (5 in our experiments) after the global maximum is detected.

beginning of the histogram. In our experiments we use $T_b = 5$. The eye region square side length is defined as $1.6 \times T_d$.

3.2 Motion Vectors

The eye region is divided into 3×3 cells (Fig. 3). An average ("cell") motion vector is calculated for a cell from the individual local motion vectors belonging to the cell based on their locations. Eye blink causes a significant vertical move in the cells of middle row, but only a minor move in the top or bottom row. Motion vectors have different characteristics during head movements or other facial mimics. The vertical components of the middle and top rows are sufficient for further computation. From these 6 motion vectors the variance $var(y)$ (Eq. 1) is calculated. The vertical component is sufficient because there is a strong predisposition that the person's face does not rotate significantly.

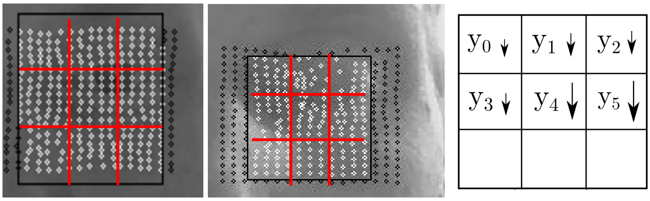

Fig. 3. The eye region with flock of KLT trackers shown before and during an eyelid moves down. The region is divided into 3×3 cells. For the given cell the average tracker locations are calculated. Vertical components of the first 6 motion vectors are sufficient for further computation. The gray dots represent trackers which are used to compute motion vectors.

$$\mu = \frac{\sum_i y_i}{6}, \qquad var(y) = \frac{\sum_i (\mu - y_i)^2}{6} \tag{1}$$

Statistical variance of the 6 upper cells represents the diversity across moves. If the variance is higher than the variance threshold T_v, it will indicate an eyelid 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 $T_v = \kappa \times \frac{d}{fps}$, where:

- κ: Based on the tests on our dataset, the constant value is 0.02.
- d: The interocular distance is directly proportional to the subject distance from the camera. The eye size affects the size of motion vectors.
- fps: The frame rate of the input video sequence also influences the size of motion vectors. Higher frame rate means lower variance of motion vectors.

Motion vectors represent local move in the eye region which is a sign of an eye blink or possibly other moves as facial mimics, eyebrow or pupil. To eliminate detection of non blink moves a simple state machine is setup (Fig. 4). We consider eye blink as the combination of two movements – down and up move. After move down detection, the state changes from *State 0* (the initial one) into *State 1*. If the state machine is in *State 1* for about 100ms (3 frames while 30fps) and then move up is detected, the state will change into *State 2* – blink occurred. If there is no move for more than 150ms (5 frames while 30fps), the state machine will change from *State 1* back to *State 0*. We follow the assumption that an average eye blink takes from 150 to 300ms.

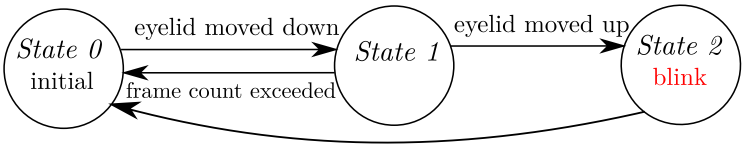

Fig. 4. The state machine of eye blink detection. After move down detection, the state changes from *State 0* into *State 1*. While in *State 1* and move up detection, the state changes into *State 2* – blink occurred. *State 1* changes back to *State 0* if there is no move for given amount of time.

Different sizes of the vertical components of the motion vectors across cells are shown in Fig. 5. Based on our observation we define move down as $(y_4 > 0)$ & $(y_4 > y_1)$ & $(var(y) > T_v)$ and move up as $(y_4 < 0)$ & $(y_4 < y_1)$ & $(var(y) > T_v)$. Algorithm 1 presents the state machine pseudo code for the eye blink detection. There are two state machines established, one for each eye. Blink state in either of the state machines defines an eye blink.

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

(a)

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

(b)

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

(c)

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

(d)

Fig. 5. Difference of vertical moves in the upper 6 cells ($y_0 \ldots y_5$) during head and eye blink movements. y_1 is used as the reference to normalize the sample data only for better visualization: (a) head moves down, all motion vectors have similar size, (b) eyelid moves down, the middle row has bigger positive change, (c) head moves up, all motion vectors have again similar size, (d) eyelid moves up, the middle row has bigger negative change. Most of the time y_1 captures the head move and y_4 the eyelid move, due to which they help to define eyelid down and up move.

3.3 Reinitialization

A uniform distribution of flock of trackers is negatively affected (Fig. 6) by motion in general. A uniform distribution is important to acquire representative data of average motion vectors. We reinitialize our algorithm with face and eye detection in case of following events:

Alg. 1. State machine detecting eye blink based on variance of vertical component of motion vectors

INPUT: 6 vertical components of motion vectors(y), current state of the state machine (state), distance between eye centers (d), fps
OUTPUT: eye blink state

```
procedure GET_EYE_BLINK_STATUS(y, state, d, fps)
    T_v ← 0.02 * d/fps                                         ▷ variance threshold
    if state = 2 then
        state ← 0
    end if
    if state = 0 then
        if ((y_4 > 0) & (y_4 > y_1) & (var(y) > T_v)) then      ▷ move down detected
            state ← 1
            time = current_time()
        end if
    else if (state = 1) & (100ms < (current_time() − time)) then
        if ((current_time() − time) < 150ms) then
            if ((y_4 < 0) & (y_4 < y_1) & (var(y) > T_v)) then  ▷ move up detected
                state ← 2                                       ▷ eye blink occurred
            end if
        else
            state ← 0                                           ▷ eye blink did not occur
        end if
    end if
    return state
end procedure
```

- after blink occurred,
- large number of lost trackers: more than half of the trackers are lost between two frames or the remaining number of trackers is less than 20,
- over time, after constant number of frames (in our implementation every 200 frames).

Reinitialization restores parameters to their initial values. However, it is necessary to preserve the current states of the state machines and blink frames counters, which preserve the information of how many frames in a row with no movement were observed during the *State 1*. This way the reinitialization will not interrupt the eye blink detection process.

4 Evaluation

We introduce a new dataset called *Eyeblink8*, which consists of 8 videos (Fig. 8) 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

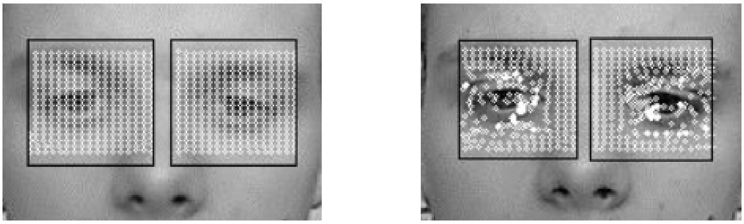

Fig. 6. Left figure shows initial flock of KLT trackers over a regular grid. Right figure represents their positions after 3 blinks. Many trackers are snapped to corners and edges disrupting the uniform distribution. Reinitialization is necessary because motion vectors rely on uniform distribution.

contains over 82600 frames (640×480) with 353 blinks. All videos are recorded using Logitech C905 camera with 30fps acquisition.

The annotation to individual videos consists of two files; the first file contains the frame numbers with the 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 eye is defined as 90–100% of the eye is covered with the eyelid. Fully closed eyes are tagged with *close* and openning eye is again tagged with *half* until it is fully open. Also not fully closed eye blinks can be annotated this way (consisting only from “halfs”). If only one eye is visible (or blinks) the tag *Left/Right* is added to the eye state, based on the location of the visible blink. Sample annotation is in Fig. 7. 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 frames interval with the detected eye blink in the middle). The intersection interval with the ground truth can be counted just once as *True Positive*.

We evaluated our method also on the 2 publicly available datasets (Talking Face video, ZJU). *Talking*[2] (Fig. 9) 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. We created a new annotation compatible with our evaluation framework described above. Results are compared with the existing methods in Table 1. We failed to detect only two blinks, which happened during the downward sight, therefore the size of the vertical component of motion vectors is not significant enough.

The ZJU dataset (Fig. 10) consists of 80 short videos (10876 frames) of 20 individuals with and without glasses (insignificant small frame) captured with 30fps 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

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

frame number	acquisition time
2036	72.5077
2037	72.5392
2038	72.5712
2039	72.6032
2040	72.6356
2041	72.6672
2042	72.6994

frame number	state
8433	half
8434	half
8435	close
8436	close
8437	half
8438	half
8439	half

frame number	state
3643	halfRight
3644	halfRight
3645	halfRight
3646	closeRight
3647	closeRight
3648	halfRight
3649	halfRight

Fig. 7. Sample annotations for the dataset *eyeblink8*

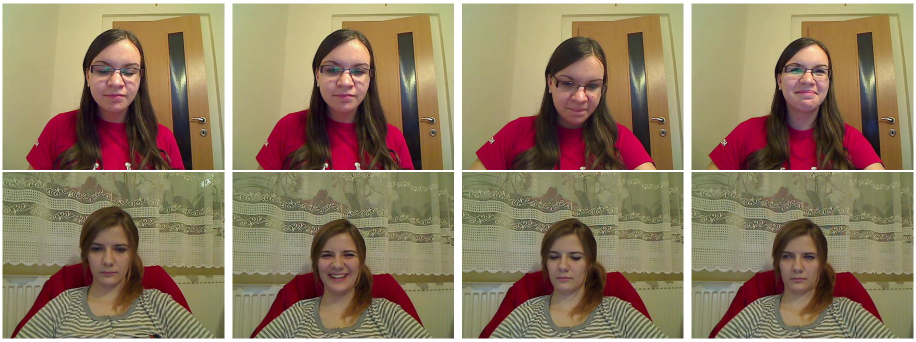

Fig. 8. Sample snapshots from our dataset the *Eyeblink8*

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. 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.

Comparison on the 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, based on the study [15], an average blink takes 150–300ms (5–10 frames while 30fps). We divided the num-

Fig. 9. Sample snapshots from the *Talking face* dataset

Fig. 10. Sample snapshots from the *ZJU* dataset

ber of frames with open eyes in datasets by the average blink duration (7 frames) and this is used as negative (number of non eye blinks). From our annotation of the ZJU dataset can be read, that 2482 frames capture some part of eye blink moves. 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, ZJU and Talking dataset. The number next to ZJU represents the number of the 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 [6]	Talking	-	95%	19%	88%
Divjak & Bischof[3]	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 [6]	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 the Talking face dataset, but lower *Recall* on the ZJU dataset. We are unable to detect 71 eye blinks on the ZJU. One third is caused by an 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

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

eye blinks are only seen in one or two frames. The fastest eye blink last at least 150ms [15] and therefore should be on 5 frames (while 30fps).

Divjak & Bischof [6] have quite high false positive rate on the Talking face video. 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 mostly because of motion blur.

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 utilization.

5 Conclusions

There is an increase attentation on eye blink detection algorithms for different purposes as driver fatigue detection or face liveness detection etc. We present a simple method based on flock of trackers and variance of motion vectors. Standard camera with 25–30fps is sufficient. By using the statistical variance of vertical component of the motion vectors as the input for the state machine, we created a robust method and achieve invariance to common head moves and facial mimics.

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 the number of non eye blinks and not the number of images containing an open eye. We achieve state-of-the-art results on *ZJU* dataset.

Acknowledgments. This work was partially supported by VEGA 1/0625/14. My thanks belongs to Matej Makula and professor Jiří Matas for the wonderful comments which helped to improve this paper.

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: 2010 International Conference on Machine and Web Intelligence (ICMWI), 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.: Real-time video-based eye blink analysis for detection of low blink-rate during computer use. In: First International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), pp. 99–107 (2008)
6. 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)
7. 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)
8. Garcia, I., Bronte, S., Bergasa, L., Almazan, J., Yebes, J.: Vision-based drowsiness detector for real driving conditions. In: 2012 IEEE Intelligent Vehicles Symposium (IV), pp. 618–623 (2012)
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. Radlak, K., Smolka, B.: Blink detection based on the weighted gradient descriptor. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierek, A. (eds.) CORES 2013. AISC, vol. 226, pp. 691–700. Springer, Heidelberg (2013)
14. Rosten, E., Drummond, T.W.: Machine learning for high-speed corner detection. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 430–443. Springer, Heidelberg (2006)
15. Stern, J.A., Walrath, L.C., Goldstein, R.: The endogenous eyeblink. Psychophysiology **21**(1), 22–33 (1984)
16. 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. IEEE ICSMC. **4**, 2891–2896 (2006)
17. Szwoch, M., Pieniążek, P.: Eye blink based detection of liveness in biometric authentication systems using conditional random fields. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) ICCVG 2012. LNCS, vol. 7594, pp. 669–676. Springer, Heidelberg (2012)
18. Tomasi, C., Kanade, T.: Detection and tracking of point features. Computer Science Department, Carnegie Mellon University, Tech. rep. (April 1991)
19. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vision **57**(2), 137–154 (2004)