Embedded Real Time Blink Detection System for Driver Fatigue Monitoring

Soheil Salehian* & Behrouz Far†
Department of Electrical and Computer Engineering,
University of Calgary,
Calgary, Canada.
Email: *ssalehia@ucalgary.ca, †far@ucalgary.ca

***Abstract*—Fatigue induced vehicle accidents have seen an increase in the last few decades. Fatigue monitoring using non-invasive and real time image processing and computer vision techniques have shown great promise and are an active research area. To that extent, in the proposed work a blink detection algorithm is proposed that serves as a visual cue that may be correlated to the state of fatigue of the driver. Using a complimentary but independent approach, shape analysis and histogram analysis are carried out in parallel to perform the blink detection task. Close to real time performance and a high level of accuracy in controlled settings show great promise of such approach in enhancing the monitoring of the driver's blinking patterns. One of the main constraints of using such algorithm in a real world setting is the minimized processing time required to allow for sufficient driver response time. In this work implementation of the algorithm is described using optimization techniques to meet such latency requirements. The validation of the algorithm was carried out by visual inspection of the video sequences in terms of precision and accuracy. The presented blink detection algorithm has a precision rate of 84% and an accuracy rate of 69% obtained through using 12 sequences of different duration videos in varying lighting conditions using a small sample of participants.**

***Index Terms*—computer vision, blink detection, driver fatigue, image processing.**

I. Introduction

The alarming number of traffic accidents due to driver fatigue accounts for more than half of all truck collisions in the United States [1]. Diminished levels of attention caused by fatigue, increases response time while in more severe cases it may result in short lapses of sleep by the driver. Research has shown that after 2-3 hours of constant driving, fatigue plays an important factor in slowing decision making and perception of the driver of the vehicle. More recently, a study by the National Sleep Foundation in the US showed that in their study, more than 51% of adult drivers with drowsy symptoms had driven a vehicle and 17% had momentarily fallen asleep while driving [2]. It is estimated that 1,200 deaths and 76,000 injuries are due to fatigue induced accidents annually. Due to the recent attention to fatigue related crashes, fatigue detection systems have become an active area of research.

There has been substantial work on characterizing driver fatigue based on various models. Dinges et al. [3] showed that physiological signals such as the electroencephalography (EEG) and the electro-cardiogram (ECG) can be used to measure fatigue. Other less intrusive methods using physical information of the vehicle combined with the patterns of driving has also been researched extensively with limited success[4]. Although the most accurate results have been reported using physiological instrumentation, such systems are not practical as they require initial setup which is a hassle for the driver [5], [6]. With non intrusiveness constraints, another category of methods that has become an active research area is non-intrusive online monitoring of the driver using computer vision. In this category, "visual cues" such as gaze, head movement, and eye blink rate is tracked in order to accurately estimate the state of the driver. These computer vision techniques aim to extract visual fatigue related characteristics in real time using image/video processing. For instance, Boverie et al. [7] developed a system to correlate eyelid movement to estimate the degree of vigilance of the driver. While others such as Ueno et al. [4] looked at methods to measure the degree of openness of the eyelids to make the fatigue characterization. The majority of such early studies involved strictly controlled environments, lighting conditions and line of sight for the extraction process to work properly. Recent clinical research on the effectiveness of blinks as strong indicators of fatigue [8], make blink detection a strong candidate that is the basis of the following work.

Although most of previous work has focused on the ability of the vision system to correctly detect blinks in various lighting conditions, in real world software vision systems there are response time requirements that dictate the effectiveness of such systems. Studies by Muttart [9] have quantitatively examined the driver response times based on various real conditions on the road. Their conclusion suggested that there is a variety in response time in drivers that is obviously correlated to driver speed and conditions. Therefore, it is of note to mention that the response time of a developed computer vision system should be minimized as much as possible to account for this variation in driver population and allow for sufficient time of reaction in real settings. From the moment the system has the image data this time is labelled *processing time* and is required to respond in less than an estimated 900ms. It is the focus of the following work to not only develop the algorithm but enhance in its performance on an implementation in an embedded system with real time constraints.

(DOI reference number: 10.18293/SEKE2015-249)

In the presented chapter, an eye blink detection algorithm is proposed using machine learning and image processing techniques in an effort to enhance the robustness of blink detection as an important part of a driver fatigue monitoring system. The contribution of this work includes two complimentary algorithms that exploit different information in each image/frame in order to arrive at a more robust estimation of the driver blink rate along with their concurrent implementation on an embedded system achieving real time requirements. The remainder of the paper is organized as follows: Section II provides a detailed explanation of the proposed algorithm methodology. The implementation details and optimization required to enhance performance of the vision system is described in Section III. The results of the critical steps of the algorithm and the validation methodology is presented in Section IV. A discussion of the results of the algorithm in a video processing setting is part of Section V and finally, conclusions on effectiveness and limitations of our approach along with future work is outlined in section VI.

II. Methods

In the following section, the eye blink algorithm is described with a detailed discussion on key sections of the algorithm such as: face detection, eye detection, pre-processing of the region of interest (ROI), shape analysis, complimentary histogram analysis method and combination of their outputs. The algorithm was designed using a set of real-time video captures in various lighting conditions for robustness verification.

A. Algorithm Overview

The high level flow of the proposed algorithm, initiates with detecting the face area using Haar-features. These features are extracted using a Haar classifier that has been trained with frontal face images. Once the face area is located, a second classifier using similar Haar-features finds the eye band area for both eyes. This eye band area is the ROI that will be processed by two separate eye blink detection methods. In one method, called pipeline A, the gray scale image of the initial frame is used as input. Then, the edges within the ROI are detected using the canny edge detector with a 3×3 Gaussian blur filter to remove noise. This edge detection works well because there is a strong contrast between the iris and the choroid.

The algorithm proceeds by doing contour analysis, where the various large contours of the ROI are examined further. Due to the elliptical shape of the eyes while being open, an ellipse fitting is performed to identify the eyes as open. Upon closure of the eyes, the number of ellipses, corresponding to each eye, in the image reduces greatly which indicates that a blink may have happened.

However our experimentation shows that contour information may not be sufficient in robustly detecting the blinks. Therefore we have chosen to implement a second computationally efficient method that concurrently enhances the detection outcome.

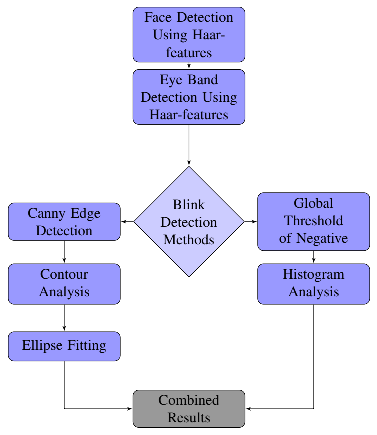


Fig. 1: Flow chart representation of the proposed blink detection algorithm. Please note the two parallel sub-algorithms and the combination of their results.

In parallel, a secondary method (pipeline B) looks at non spatial information of the ROI. This second method, takes the negative of the frame and uses a simple threshold to globally threshold high pixel values (which will include the surrounding areas around the eyes and the eye structure). The histogram of such binary image has a bimodal shape with two impulses. It can be observed that blinking reduces the number of white pixels because momentarily the eyelids will cover the eyes and there is a shift of pixels from the high end of the histogram to the low end which is detected by the algorithm. The fusion of the results of contour/shape analysis with histogram analysis allows for the detection of eye blinks.

A flowchart representation of the algorithm overview is presented in Fig. 1. The various stages of the algorithm is explained in more detail in the following sections.

B. Face Detection

The following section presents the learning-based method used in detecting the face area in our blink detection algorithm. Learning based methods use training samples in combination with statistical and machine learning models that have have shown to be effective in detecting facial features [10]. One main advantage of learning methods is their ability to adopt to various scenarios given adequate and large training sets. Variety of lighting conditions, driver demographics and other features specific to the driving of the vehicle can be included in the training set in order to increase accuracy and robustness of the face detection process.

Viola et al. [11] proposed a set of features named Haar-like features due to similarity to Haar wavelet basis functions. Their algorithm has gained popularity due its robust and computationally efficient property for object detection specifically in the face detection domain. Haar-like features use the change in contrast of adjacent rectangular groups of pixels instead of the pixel's own intensity values. The variance between the neighbourhoods surrounding the pixel are used to identify areas of high and low intensity values. Different number of grouping of such basis functions based on their variance can result in detecting different types of features such as edge, line or center-surround features [12].

The simplicity of these features allow for scaling and therefore scale-invariant detection of face region in the frame. Viola et al. [11] showed that for a rather small image, the total number of such elementary features is in the order 180,000 which may be impractical to calculate [12]. However, for accurate object detection, they noted that not all features are required. By transforming the image into what they called an ”integral image”, any of the haar features is able to be computed at any scale in constant time. The construction of the features is initiated by generating the integral image. The integral image intermediate representation (iI) of original image (I) at *x,y* contains the sum of pixels above and to the left at *x',y'* which can be formally defined as:

$$iI(x,y) = \sum_{x' \leq x, y' \leq y} I(x',y')$$

Then the cumulative row sum *s(x,y)* is:

$$s(x,y) = s(x,y-1) + i(x,y)$$

Then the integral image can be re-written in terms of the cumulative row sum as:

$$iI(x,y) = iI(x-1,y) + s(x,y)$$

Using this simple technique in generating the integral image, all the combination of rectangle feature sets may be constructed which makes feature generation computationally efficient. These features are sensitive to presence of edges, bars, and simple structures with only horizontal, vertical and diagonal orientation [11]. Specifically in the case of face detection, it was noted that some features are more effective than others based on exploiting the property of the region of the eyes that is often darker than the region of the nose and cheek (with a higher variance in the eye region) and similarly darker region of the eyes from the bridge of the nose.

The effect of different lighting conditions are important in a variance based method and therefore during the training it was addressed by a variance normalization procedure defined as:

$$\sigma_w^2 = \mu_w^2 - \frac{1}{N}\sum p_w^2$$

Where the variance σ^2 of window *w* is defined in terms of mean of the window (μ_w) and the sum of squared pixels *p* of window *w*. The summation is calculated using the integral image procedure previously described. It is important to note that such normalization in the training procedure is inherently needed in the detection phase as well.

With the feature generation phase complete, a learning method is required in order to perform a classification function. The AdaBoost learning algorithm is used for both tasks of feature selection and the training of the classifier [13]. Using the Adaboost weak learner procedure, each classifier can only depend on a single feature and cascading of such classifiers allow for a robust method for scale invariant object detection.

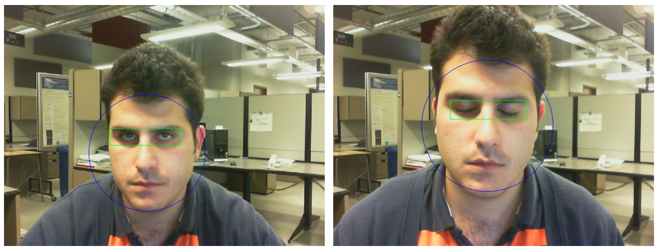

(a) Open eyes ROIs (b) Closed eyes ROIs

Fig. 2: Results of face (blue) and eye band (green) ROI detection using Haar-classifiers. The classifier has worked well in in identification of the eye ROI regions while the eyes are closed in (b).

A large reduction in the number of non-contributing features, and its excellent generalization performance allows AdaBoost to be used in a cascaded format that forms the cascade classification of haar-features. The cascading of classifiers allows for training each classifier using AdaBoost and adjusting each classifier's threshold and weights to minimize false negatives using error minimization.

The results have proven to have a high accuracy rating and a subsequent better performance as the cascading continues. This makes the haar-like cascaded classifier method ideal for the face detection task of our real time blink detection algorithm.

C. Eye Band Detection

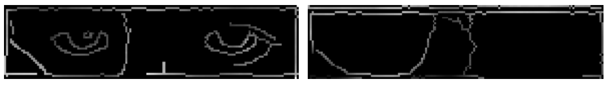
The eye band detection method to identify the eyes in each frame uses the same methodology as the face detection mechanism via Haar-like features and AdaBoost combination. Beyond the technical aspects of the classifier mechanism, for the eye detection few considerations are worth mentioning:

- By finding the face region in each frame, the ROI for eye detection becomes smaller and the performance of the classifier increases dramatically.
- A separate training set for eyes is used to train the classifier. In the case of this work in its current form, a pre-trained classifier was used which performed adequately for the eye detection task.
- The decision was made to detect both eyes as an ”eye band” as the trained classifier performed best when both eyes were facing the camera.

Fig. 2 demonstrates the results of simultaneous face and eye detection for frames corresponding to both open and closed eyes using the described procedure.

D. Canny Edge Detection

Edge detection is an important procedure and a first step in identifying objects of interest in the image. Once the ROI has been identified, edge detection allows for structural analysis



(a) Open eyes Canny output (b) Closed eyes Canny output

Fig. 3: Results of canny edge detection using $lowThreshold = 60$ and $maxThreshold = 140$ for both open and closed cases. Please note that while closed, the detector has only detected the shadow area around the eyes.

inside the ROI which in the case of this work is the eye band detected in the previous stage. There are a number of edge detectors that may be used depending on the desired structural properties. In the proposed algorithm, the popular Canny edge detector algorithm [14] was selected due to its following characteristics:

1) Robust detection: in the blink detection application, the probability of detecting real edges need to be quite high despite high noise levels in each frame.
2) Computationally inexpensive: due to the real time nature of the application, high performance was a secondary but important deciding factor.
3) Step edges: the strong variance between the eye region and skin (both horizontally and vertically) can be characterized as step edges which the canny algorithm was originally designed for [14].

The resulting output of this stage, is a binary image that identifies all the corresponding edges in the eye band area. Fig. 3 shows the effect of the operator in open and closure sample cases described previously. It is worth mentioning that the low threshold for the method was proven to be critical in detecting important edge structure between the eye choroid and the pupil. A severely low threshold would mark many details of the eye band as edges which was sub-optimal for the blink detection algorithm. The thresholds were chosen empirically based on the training set frames during development.

E. Contour Extraction & Analysis

By finding the strong edges using the previous operation in the eye band region, the structural information of the ROI is ready for further analysis in detecting the eye regions. The goal of this proposed stage is to find an approximation of all the contours that are present in each frame. The following section explains in detail how contour extraction from the edge information is accomplished and the corresponding analysis that is carried out on each contour.

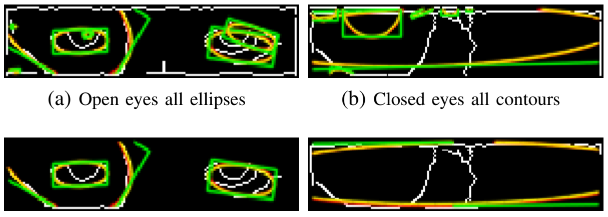
The contour extraction operation used in our algorithm is based on the work of Suzuki et al. [15] where a topological analysis is done on the contours found by what is known as ”border tracing” based on earlier work by Rosenfeld et al.[16] and the utilization of Freeman's chain codes [17].

Once all contours are traced using the above algorithm, the operation proposed by Suzuki et al. derives a sequence of coordinates on each contour and constructs a topological ordering of such coordinates. It was shown that using such technique, both outer contours and inner contours (holes) can be effectively labeled and the topological analysis can lend to accurate categorization and discrimination of enclosing contours vs. inner contours. In the case of our blink detection algorithm, due to the large size of both eye regions in the band area, it was desired to analyze the outer contours in each frame. The proposed algorithm exploits the fact that during the blinking motion, larger contours of the eye will be deformed and disappear rapidly and hence extracting the contours is a first step in monitoring the blinking. Once the contours are extracted, the proposed algorithm proceeds by calculating the area of each contour for further analysis. It was observed that discriminating the large contours (such as large reflections due to sever lighting conditions) or smaller extracted contours (due to poor edge extraction) based on area threshold was an effective method in keeping only contours related to the eye region. The area threshold is adaptive and based on the size and resolution of the eye band frame.

F. Ellipse Fitting

Once the corresponding contours to the eye region (the choroid and iris sections) have been identified, shape analysis is the next stage of the proposed algorithm. In this section, the ellipse fitting procedure and some of the assumptions and criteria of the analysis is discussed in more detail. The intuition behind the procedure is based on a *priori* that the eye region contours have an elliptical shape. Fitzbibbon et al. [18] evaluated various methods of fitting data to conic sections based on assumptions of isotropic normally distributed noise and incomplete contours. Their work is of particular interest in our application due to its analysis of performance in terms of algorithm complexity and computation in the task of ellipse fitting using least-squares as a distance metric. Fitzbibbon et al. showed that based on their experimentation evaluation of the popular conic fitting algorithms with strong variants of noise, orientation, and occlusion, that least-squares based algorithm of statistical distance (also known as BIAS [19]) has a good tradeoff between performance and accuracy in fitting contour points to an ellipse even with the presence of outliers due to high noise and discontinuities.

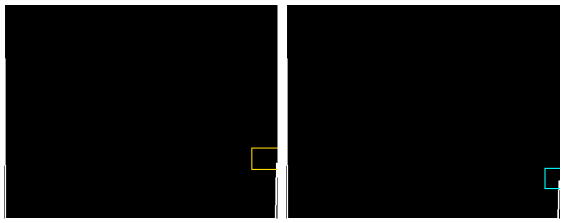
During the shape analysis stage of the blink detection algorithm (implementation in OpenCV), all fitted ellipses had to meet a discrimination criteria to be included for further stages. The orientation of the fitted ellipses become important in distinguishing eyes from other ellipses. It can be observed that contours of the eye region has an orientation close to the horizontal line with some varying angle, θ, assuming that the driver is not orienting their head substantially. Choosing an appropriate threshold on θ allows the algorithm to only include ellipses resembling the contours of the eye region (θ was empirically found to be $10°$). Another discriminant which was used to avoid fitting all possible contours was the contour area. Fig. 4 shows the results of ellipse fitting with and without constraints in both open and eyes closed cases. Please note that the fitted small ellipse in Fig. 4 (b) has been discarded using a contour area threshold based on the resolution of the frame



(a) Open eyes all ellipses (b) Closed eyes all contours

(c) Open eyes all ellipses (d) Closed eyes all contours

Fig. 4: Results of the ellipse fitting procedure using discriminant of $\theta = 10^\circ$ and $contourArea = 20$ pixels. The red and yellow line are the ellipse fitting procedure while the green box is the rectangular fit for orientation analysis.



(a) Open eyes global threshold (b) Closed eyes global threshold

Fig. 5: Results of global thresholding, $\tau = 185$ on the negative of frames for both open and closed cases. Please note the decrease of the number high intensity pixels in (b).

in Fig. 4 (d). The effect of orientation discriminant can be observed in between Fig. 4 (a) and (c).

With the shape analysis in place, this pipeline of the algorithm (pipelline A) is able to classify the eyes in the frame as open or closed. During the blinking motion, the monitored number of allowable ellipses at each frame has a reduction of 2 or more which will indicate that the eyes have closed and therefore the frame can be marked accordingly.

Although this pipeline works in most test cases, it was observed that due to variations of sampling from different experimented cameras and the variability in lighting conditions, the pipeline requires a complementary synchronization mechanism to overcome some of these shortcomings. A second complimentary pipeline (pipeline B) was developed to run in parallel which will be discussed in the following section.

G. *Global Thresholding Of Negative*

To independently complement the shape/contour analysis pipeline discussed in the previous sections, pipeline B was developed using a rather simple thresholding technique. It was observed that during the blinking motion, when the eyelids cover the choroid and iris there is a change in the number of pixels that represent the skin. By exploiting this idea, a global threshold on the negative of the gray scale frame is used to detect the eye region and its approximate surroundings. The global thresholding is formally defined as [17]:

$$g(m,n) = \begin{cases} 0 & \text{if} f(m,n) \leq \tau \\ 255 & \text{if} f(m,n) > \tau \end{cases}$$

where the resulting binary image, $g(m,n)$, is based on the global thresholding operation on the original gray scale image $f(m,n)$ with threshold τ. This procedure renders robust results with prior knowledge of the lighting conditions and range of skin pixel values.

H. *Histogram Analysis*

Using the resulting binary image, simple histogram analysis is used to monitor two groups of pixels in the image. One group belongs to the background and the other group is the approximate eye region. During the blinking motion, there is a substantial change in the number of high intensity pixels and a comparison of this number with the previous frame has shown to be robust in the preliminary results of this work. Using this difference, $\%d$, the frame is classified as closed only if the difference is greater and equal to 20% of the previous frame's number of high intensity pixels (found empirically). Fig. 5 shows the sudden drop of the number of high intensity pixels during the blinking motion.

I. *Merging of Results*

In this final stage of the algorithm, the results of the shape analysis/elliptical fitting (pipeline A) and the results of global thresholding (pipeline B) is merged for each frame. If results from both pipelines match to be closure, the frame is classified as a detected blink. If there is a discrepancy among the pipelines, the result is marked as an open eye.

III. Real-time Implementation

In this section some of the details of the implementation of the proposed algorithm is presented. As previously mentioned, strict real time requirements are present in order for the system to fulfill the minimization of *processing time* of 900 ms and allow for slowest human driver response time in the process. The embedded system chosen for this work was the Nvidia Jetson TK1 which includes an ARM Cortex A-15 dual core processor and a Nvidia GPU for embedded vision applications on the Ubuntu 14.04 L4T platform. Although our development environment was similar on Linux 14.04 on x64 machine, during the porting process on the target embedded system, there were optimization to be considered. With the latency constraint of *processing time* being 900 ms it was important to optimize CPU code.

One of the main steps in the algorithm that is used by both pipeline A and pipeline B extensively was the color to gray scale conversion. With the target platform being an ARM based CPU with ARM's NEON [20] capabilities which allows for single instruction multiple data (SIMD) operations, the approach chosen was to enhance gray scale conversions using NEON intrinsic instructions.

The optimized implementation of gray scale conversion using NEON intrinsics is shown below:

```
void neon_rbg_gray (uint8_t * __restrict dest,
uint8_t * __restrict src, int numPixels)
{
```

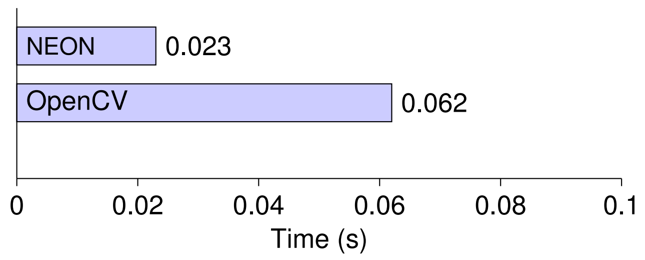


Fig. 6: Comparions of our Neon optimized RGB to gray scale conversion compared to native OpenCV C++ performance.

```
  int i;
  // 8x8 Neon registers are filled
  // Red channel multiplier
  uint8x8_t rfac = vdup_n_u8 (77);
  // Blue channel multiplier
  uint8x8_t gfac = vdup_n_u8 (151);
  // Green channel multiplier
  uint8x8_t bfac = vdup_n_u8 (28);
  int n = numPixels / 8;

  // Conversion in 8 pixel chunks
  for (i=0; i < n; ++i)
  {
    uint16x8_t  temp;
    uint8x8x4_t rgb  = vld4_u8 (src);
    uint8x8_t result;

    temp = vmull_u8 (rgb.val[0],      bfac);
    temp = vmlal_u8 (temp,rgb.val[1], gfac);
    temp = vmlal_u8 (temp,rgb.val[2], rfac);

    result = vshrn_n_u16 (temp, 8);
    vst1_u8 (dest, result);
    src  += 8*4;
    dest += 8;
  }
}
```

Fig 6 shows some of the earliest results with comparison to OpenCV's native C++ implementation. As it can be observed a reduced factor of 2.5x allows for the system to meet the latency requirements of *processing time* of 900ms.

IV. Results

The following section outlines the results obtained through experiments of this work. The proposed algorithm was developed and visually validated using live video processing of a 720p webcam at 20 frames per second using the OpenCV C++ APIs. For validation of this work, our preliminary validation procedure was carried out from a set of sequence of frames picked from various recordings in different lighting conditions and a small sample of subjects. The sampling included a relatively even distribution of the three possible cases of: open eyes, closure motion, and closed eyes. As it is evident, the algorithm initially needs to detect the face and eyeband ROI region accurately and reliably prior to moving on to the later stages for blink detection. By constraining the distance away from the camera and the head orientation of the driver, it was found that the Haar-feature classifier performs well based on our qualitative analysis of frames through out development and on the sequence frames used for validation.

The distribution of the sequences which can be observed in Table I, are as follows: sequences $s1 - s5$ were based on moderate illumination conditions, $s6 - s9$ were realized in highly illuminated and $s10 - s12$ were in low illuminated conditions. The results including the accuracy and precision of each sequence have been based on the assumption that the eyeband ROI detection procedure has been successful (as the algorithm will not carry on if the eye band ROI is not detected). True positives (TP) include all cases that both eyes were closed and the system was able to detect the blink. The cases of detected false blinks when they were either open or the eye band ROI was not even detected, is labeled as false positives (FP). The percentage of inability of the algorithm to detect closure while detecting the eyeband ROI region is included under the false negatives category (FN). The number of actual positives (AP) was measured by repeated review of each sequence via only visual inspection at this time.

TABLE I: Results of proposed blink detection algorithm in different sequences of frames in moderate ($s1-s5$), high ($s6-s9$) and low illumination ($s11-$s12) conditions. Accuracy and precision of each sequence and the total average has been shown.

Sequence #	Frames	AP	TP	FP	FN	Accuracy	Precision
s1	242	16	15	1	0	0.9375	0.9375
s2	200	12	12	0	0	1.0000	1.0000
s3	193	8	8	0	1	0.8889	1.0000
s4	300	4	3	1	2	0.5000	0.7500
s5	275	21	19	2	1	0.8634	0.9048
s6	250	14	12	2	9	0.5218	0.8571
s7	220	9	8	1	6	0.5333	0.8889
s8	244	11	9	2	8	0.4737	0.8182
s9	207	7	5	1	1	0.7142	0.7143
s10	190	10	6	4	2	0.5000	0.6000
s11	202	12	9	3	2	0.6428	0.7500
s12	248	14	13	1	3	0.7647	0.9286
Average						**0.6942**	**0.8458**

A noteworthy detail of the implementation of the algorithm is that if for any reason the eye ROI is not found, the algorithm stays idle without exiting the program. The detection resumes as normal once the ROI is found, which helps in eliminating non relative frames in the validation procedure. However as it was discussed previously, in almost all cases 100% of frames were included due to both the highly robust performance of the face/eye detection stages and also the controls of the experiments.

V. Discussion

The preliminary results of the validation of the algorithm, show promise of the proposed complimentary approach of using shape analysis in parallel with histogram analysis. This is apparent in the accuracy and precision results in sequences $s1 - s5$ of Table I, where the small sample of participants in moderate and consistent lighting conditions. However, it was observed that both the canny edge detector and global thresholding methods are sensitive to both highly illuminated environments (observed in sequences $s6 - s8$) and low illumination conditions (sequence $s10 - s11$).

Specifically, the accuracy of sequences $s6-s8$ has decreased substantially in comparison to the moderate lighting conditions of the first set of sequences. The analysis shows that this might be due to the performance of the canny edge detector. It was observed that the low threshold and maximum threshold during its hysteresis phase, needed to be adjusted manually in order to improve the performance of the edge detection in low illumination conditions. To improve on this shortcoming, histogram equalization of the original gray scale was applied which has helped in reducing low illumination effects. The preliminary results of inclusion of the histogram equalization operator is shown in sequence $s12$ for reference with improvements in accuracy with a slight negative effect on precision. Further validation is required to concretely conclude the effect of the operator.

For high illumination cases, the histogram threshold had to be re-adjusted as it was observed that due to high illumination around the eyes, the thresholding results would include a larger area in the proximity of the eyes and hence a higher number of high illuminated pixels. This higher number effects the histogram analysis which means that the difference between the open eye frame and a closed eye frame may be smaller than the predefined difference $\%d$ set in average illumination cases. The results show degradation of performance in sequences $s10 - s11$. To improve on the robustness of choosing $\%d$, a normalization factor may be used. The ratio of the number of high pixels to low pixels seem to have eliminated this issue in controlled conditions resulting in performance improvements especially to precision as it can be seen in the results for $s9$. However extensive validation is required to analyze the full effect of this correction.

In terms of computional performance, our implementation allows to meet the minimum driver response time requirements of the real time system the algorithm is designed for. More extensive profiling shows other areas of improvements such as using the GPU to enhance the performance of edge detection using the Canny edge detector.

VI. Conclusion & Future Work

This work presented a blink detection algorithm based on two complimentary, but independent approaches using shape and histogram analysis. The monitoring of the driver's blink patterns was performed in near real time using efficient image and computer vision techniques. The preliminary results in terms of total accuracy and precision rates indicate that the current approach can be useful in monitoring blink detection for fatigue. The algorithm requires training in more varying lighting conditions in order to be robust. Future work will include improvements to the image acquisition system such as using an infra red camera and also additional preprocessing techniques such as gamma correction or histogram equalization. Furthermore the inclusion of adaptive methods in the edge detection step and also in the global thresholding stages will be part of the continuation of this work. Using similar techniques in identifying other visual cues such as facial expressions and yawning may enhance the accuracy of a well defined driver fatigue detection in the future.

Acknowledgment

This work has been supported by the Alberta Innovates Technology Futures and the University of Calgary. The authors would like to thank Dr. Rangayyan for the technical discussions during this work.

References

[1] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *Vehicular Technology, IEEE Transactions on*, vol. 53, no. 4, pp. 1052–1068, 2004.

[2] W. Wierwille, "Overview of research on driver drowsiness definition and driver drowsiness detection," in *Proceedings: International Technical Conference on the Enhanced Safety of Vehicles*, vol. 1995. National Highway Traffic Safety Administration, 1995, pp. 462–468.

[3] D. Dinges and M. Mallis, "Managing fatigue by drowsiness detection: Can technological promises be realized?" in *International Conference On Fatigue and Transportation, 3RD, 1998, Frementle, Western Australia*, 1998.

[4] H. Ueno, M. Kaneda, and M. Tsukino, "Development of drowsiness detection system," in *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*. IEEE, 1994, pp. 15–20.

[5] M. Kaneda, H. Iizuka, H. Ueno, M. Hiramatsu, M. Taguchi, and M. Tsukino, "Development of a drowsiness warning system," in *Proceedings: International Technical Conference on the Enhanced Safety of Vehicles*, vol. 1995. National Highway Traffic Safety Administration, 1995, pp. 469–476.

[6] S. Saito, "Does fatigue exist in a quantitative measurement of eye movements?" *Ergonomics*, vol. 35, no. 5-6, pp. 607–615, 1992.

[7] S. Boverie, A. Giralt, J. Lequellec, and A. Hirl, "Intelligent system for video monitoring of vehicle cockpit," SAE Technical Paper, Tech. Rep., 1998.

[8] R. Schleicher, N. Galley, S. Briest, and L. Galley, "Blinks and saccades as indicators of fatigue in sleepiness warnings: looking tired?" *Ergonomics*, vol. 51, no. 7, pp. 982–1010, 2008.

[9] J. W. Muttart, "Quantifying driver response times based upon research and real life data," in *3rd International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*, vol. 3, 2005, pp. 8–29.

[10] A. A. Lenskiy and J.-S. Lee, "Drivers eye blinking detection using novel color and texture segmentation algorithms," *International Journal of Control, Automation and Systems*, vol. 10, no. 2, pp. 317–327, 2012.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–511.

[12] P. I. Wilson and J. Fernandez, "Facial feature detection using haar classifiers," *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127–133, 2006.

[13] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[14] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[15] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.

[16] A. Rosenfeld and A. C. Kak, *Digital picture processing*. Elsevier, 1982, vol. 2.

[17] R. M. Rangayyan, *Biomedical image analysis*. CRC press, 2004.

[18] A. W. Fitzgibbon, R. B. Fisher *et al.*, "A buyer's guide to conic fitting," *DAI Research paper*, 1996.

[19] K.-i. Kanatani, "Statistical bias of conic fitting and renormalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 320–326, 1994.

[20] C. Pujara, A. Modi, G. Sandeep, S. Inamdar, D. Kolavil, and V. Tholath, "H. 264 video decoder optimization on arm cortex-a8 with neon," in *India Conference (INDICON), 2009 Annual IEEE*. IEEE, 2009, pp. 1–4.