2008 IEEE Intelligent Vehicles Symposium
Eindhoven University of Technology
Eindhoven, The Netherlands, June 4-6, 2008

A Computer Vision-Based System for Real-Time Detection of Sleep Onset in Fatigued Drivers

Alexandra Branzan Albu, Ben Widsten, Tiange Wang, Julie Lan, and Jordana Mah
Dept. of Electrical and Computer Engineering, University of Victoria (BC), Canada
aalbu@ece.uvic.ca, bwidsten@uvic.ca, tiangew@uvic.ca, julielan@uvic.ca, jordana@uvic.ca

Abstract - **This paper proposes a novel approach for the real-time detection of sleep onset in fatigued drivers. Sleep onset is the most critical consequence of fatigued driving, as shown by statistics of fatigue-related crashes. Therefore, unlike previous related work, we separate the issue of sleep onset from the global analysis of the physiological state of fatigue. This allows us for formulating our approach as an event-detection problem. Real-time performance is achieved by focusing on a single visual cue (i.e. eye-state), and by a custom-designed template-matching algorithm for on-line eye-state detection.**

I. Introduction

According to recent road safety surveys, fatigued driving is a common problem amongst Canadian drivers. Beirness et al [1] reported that 20% of drivers in Canada (approximately 4.1 million drivers) admitted to falling asleep or nodding off at least once while driving in the past 12 months preceding this survey. The main concern related to fatigued drivers falling asleep is their high crash rate, as well as the type of crashes that they are most likely to get involved in. As shown in Horne [2], up to 20% of serious crashes in US are sleep-related; such crashes occur during both night-time and day-time.

Fatigue and sleepiness are typically reflected in a person's facial expression and affect eye-lid movement, gaze orientation, and head movement. Such visual cues can be exploited by computer vision techniques for the detection of the fatigue and vigilance levels in drivers. This class of techniques offers a promising, non-invasive and low-cost alternative to physiological, electrode-based measures of fatigue such as those described by Heitmann et al in [3].

The majority of computer vision approaches for monitoring fatigued driving rely on special hardware. For instance, Ji and Yang [4] monitor driver vigilance with a custom-designed hardware system for real-time image acquisition and illumination control; the system is coupled with various computer vision algorithms for eye tracking and facial pose estimation. Zhu and Ji [5] use remotely located CCD cameras equipped with active infrared illuminators; they propose a probabilistic model for predicting fatigue by combining the observed visual cues (eye and head motion) with contextual information (sleep history, time of the day, temperature etc.). Active infrared vision is also used by Bergasa et al [6], who monitor driver vigilance by detecting and tracking eye-pupils (represented by small bright circular spots) on a frame-by-frame basis.

The approach proposed by Smith et al [7] [8] tracks eye and head motion from video data acquired with one camera placed on the dashboard of the car. In terms of simplicity of the hardware architecture, it is similar to the approach described in this paper. However, their scope lies in monitoring visual attention, which is different from driver fatigue; decreasing visual attention might happen when the driver is fully awake, yet not fully concentrated on the driving task. Furthermore, the framework in [9] is not designed for event-detection purposes; instead, it collects driver information relevant for visual attention in order to enable future inferences made by human experts about the driver's alertness.

The work proposed in this paper is related to driver fatigue monitoring. The novelty in our approach with respect to related work lies in its design as a real-time event detection (i.e. sleep onset) technique. Therefore, our approach does not need a complex modeling of the driver's mental state as in [5][8]. Furthermore, we show that tracking in real-time of a single visual cue (closed versus open eye state) yields good event detection performances.

The remainder of this paper is organized as follows. Section 2 describes the proposed approach. Section 3 shows experimental results, while section 4 draws conclusions and describes future work.

II. Proposed Approach

The flowchart in Figure 1 shows an overview of the proposed approach. The proposed algorithm is designed for on-line data processing; thus, it reaches the decision whether or not to activate a warning audio alarm with a latency which is lower than the inverse of the frame rate of the video acquisition process (typically ranging from 24 to 30 fps). The main steps of the algorithm are detailed below.

A. Initialization of video capture

The web-cam is connected with a video processing unit (i.e. a laptop for this pilot study) and positioned so that it acquires facial images of the driver. Therefore, it is assumed that the eye plane is orthogonal to the optical axis of the webcam (see Figure 10 for examples) and that the eyes' image is located in the central part of the frame. Our approach tolerates some variability in the distance from the eye plane to the webcam, since it can detect eye-states from very close as well as more distant views.

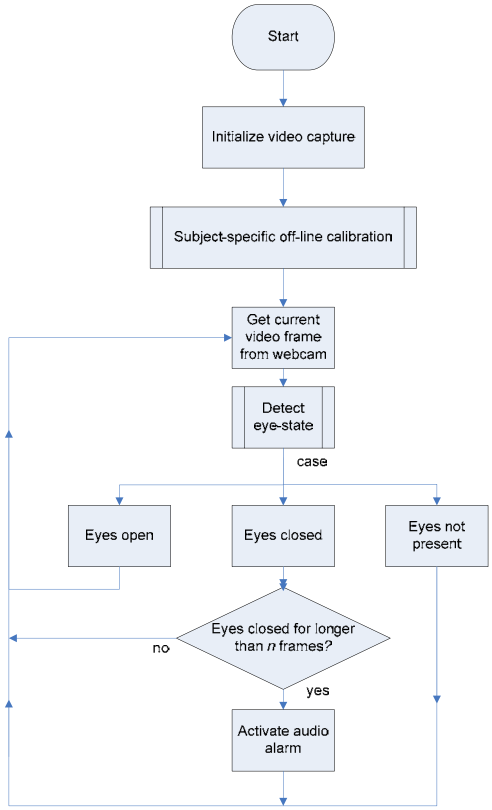


Figure 1. Flowchart of the proposed approach; modules about calibration and eye-state detection are detailed in sections B and C.

B. *Calibration*

Calibration is subject-dependent and is performed off-line, prior to eye-state monitoring. Calibration serves two purposes, namely acquisition of templates for eye-states and parametric set-up (see section C for the parameters used by the proposed approach).

For a given subject facing the webcam, the calibration process acquires two user-specified visual templates (one for eyes-closed and one for eyes-open) via a graphical user interface. This interface displays the input video-stream and a superimposed transparent, black-lined calibration box (see Figure 2). The user positions his/her eyes inside the calibration box and 'freezes' each template via a simple key-press. The flowchart of the calibration process is shown in Figure 3. The templates generated by calibration are to be used for the on-line eye-state detection described in *C*.

Figure 2. Driver positions her eyes in the calibration box, in preparation for 'freezing' the open and closed eye-state templates.

The proposed calibration approach provides a simple solution to the issue of variable lighting conditions in real-time feature recognition. Indeed, since the templates are acquired in quasi-identical lighting as for the immediately following eye-state monitoring process, the eye-state detection does not need to compensate for variable lighting. If ambient lighting conditions change significantly during eye-state monitoring with respect to calibration (for instance rapidly decreasing daylight at sunset), then calibration needs to be redone. Moreover, as shown in Figure 2, the user does not need to precisely center his/her eyes within the calibration box; template matching functions well with non-centered images, as long as the head is not tilted sideways (which is not a typically encountered head motion in driving).

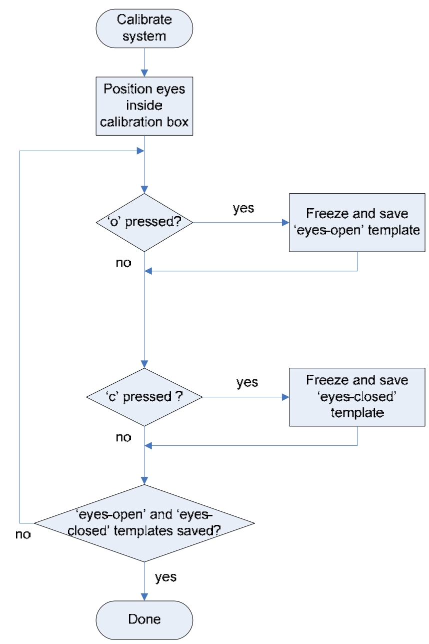


Figure 3. Flowchart of calibration process

C. *Eye-state detection*

The detection of the two eye-states is performed using a customized template matching technique on a frame-by-frame basis. As shown by Jain et al [9], template matching works well only when templates are not distorted during the imaging process; this condition is verified by the task at hand. The rationale for using this type of pattern recognition is based on the fact that, while the driver faces the road ahead, the eye pattern is not distorted; while driving, humans maintain most of the time a quasi-constant distance from the windshield. Temporary, short duration eyes-off-the-road side or bottom glances are performed only when the driver is awake; thus, they are all classified by our approach as 'eyes not present' cases which does not trigger false alarms (see Figure 1).

Template matching is typically computationally intensive [9]. In order to accurately detect matches for eyes open and closed in real time, the proposed approach offers a new customized version of template matching. The main elements that differentiate our approach from standard template matching are discussed below.

1) Defining a search region for template matching

The search region for the match detection is limited to a central zone of the currently analyzed frame (see Figure 4), which occupies a user-defined percentage of the total frame area. The value of this parameter is set up during the off-line calibration; the default value is 60%.

2) Match detection

At a given frame location, the shifted template and the rectangular frame region overlapped by it are first compared on a pixel-by-pixel basis. For each pixel location (i,j) i=1..m, j=1..n the similarity ratio $\tau(i,j)$ is computed as follows :

$$\begin{aligned} R_W(i,j) &= \tau_R(i,j) R_T(i,j); \\ G_W(i,j) &= \tau_G(i,j) G_T(i,j); \\ B_W(i,j) &= \tau_B(i,j) B_T(i,j); \\ \tau(i,j) &= \min(\tau_R, \tau_G, \tau_B); \end{aligned} \quad (1)$$

where:

- m, n are the dimensions of the template;
- R, G, and B are intensities of red, green, and blue color components;
- W and T indexes stand for the overlapped frame window and the template respectively;
- τ_R, τ_G, and τ_B are colour-specific similarity ratios.

At a pixel level, it is considered that two pixels located at (i,j) in the frame window and in the shifted template respectively are matched if their similarity ratio exceeds a certain threshold:

$$\tau(i,j) \geq \tau_0 \quad (2)$$

where τ_0 is the tolerance at the pixel level and can be user-specified during the off-line calibration step. The default value is τ_0=80%.

The global similarity between the overlapped frame window and the shifted template is computed as the percentage of matched pixels versus the template size:

$$\tau_g = \frac{card\{(i,j) \in W | \tau(i,j) \geq \tau_0\}}{m \times n} \quad (3)$$

It is considered that the current window matches the template when the global similarity exceeds a threshold:

$$\tau_g(W) \geq \tau_R \quad (4)$$

where τ_R is the tolerance at the region level and can be user-specified during the off-line calibration step. The default value is τ_R=80%.

3) Scanning strategy

Scanning for template matching is performed by iterative shifts of the template inside the search region. The iteration scheme (see Figure 4) is designed to that it decreases the number of vertical shifts necessary to detect the match. Specifically, the search starts on the median row of the search region; this starting point is consistent with the initialization of the video capture in *A*. If no match is found on this row, the next two iterations search for matches on rows located at the same distance *y* above and below the median row respectively. The distance *y* (in pixels) increases with the index *n* of the iteration according to:

$$y(n\text{-}1) = y(n) + (\text{-}1)^n \cdot n \quad (5)$$

On each row, horizontal scanning is performed from left to right; the template is shifted 1 pixel right at each iteration. Figure 4 shows three non-successive iterations from horizontal scanning performed on a given row.

Figure 4. Iterations of horizontal scanning for pixel matching

The choice of different scanning techniques along horizontal and vertical directions is motivated by the difference in vertical and horizontal head movements while driving. A driver often uses horizontal head movements (left-to-right or vice versa) in order to adjust his/her field of view to events of interest on the road. On the other side, vertical head motion is rarely used, since it does not contribute to a useful adjustment of the field of view.

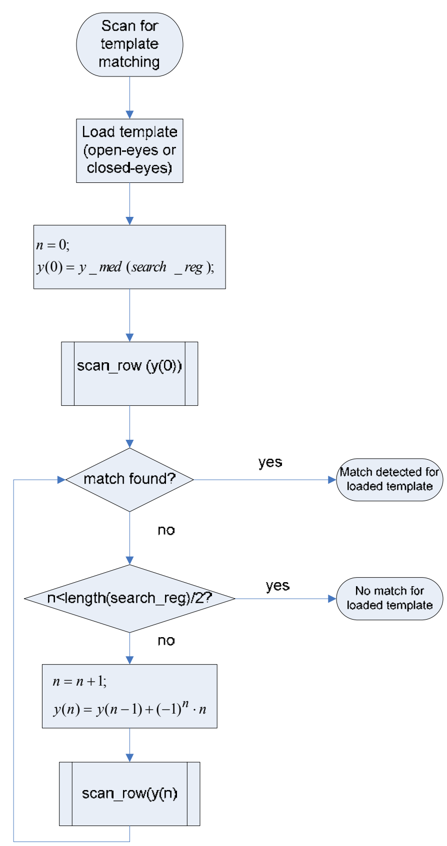


Figure 5. Flowchart of scanning algorithm

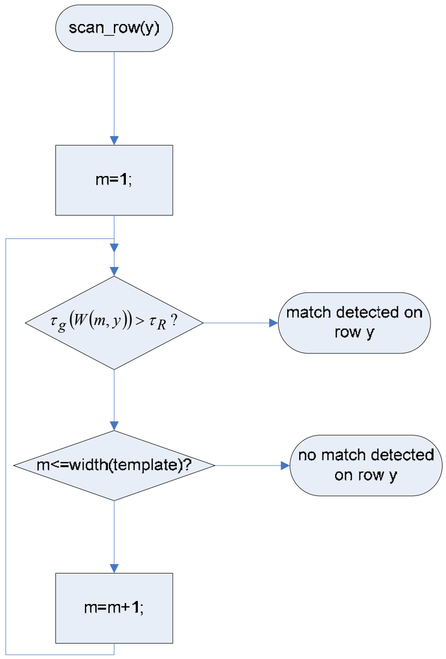


Figure 6. Flowchart of horizontal scanning

4) *Non-exhaustive search*

In a given frame, template matching is not performed with an exhaustive search. In order to increase the speed of computation, the first location of the match where the similarity ratio τ exceeds the tolerance τ_R is retained.

5) *Sequencing two template matching processes*

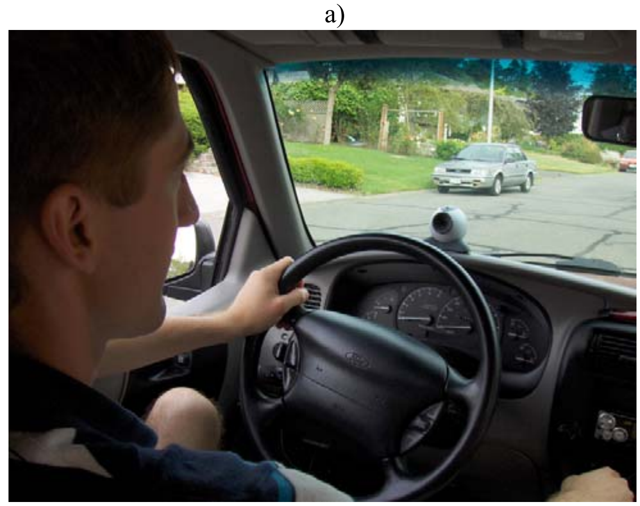
Since open-eye states are much more frequent than closed-eye ones, template matching for open-eyes is performed first. In case no open-eyes match is detected, a matching process against closed-eyes is performed afterwards. The decision of no match is reached if both matching processes do not detect matches.

III. Experimental results

The hardware used in the current implementation of the proposed approach consists in a webcam connected to a laptop. Inexpensive hardware was deliberately chosen in order to demonstrate that the prototype system can function with low cost components and therefore could be affordable to a large segment of the driving population.

Experimental results were acquired using as webcam the Logitech QuickCam® Communicate STX™, and as laptop the Dell Inspiron 6400 1.6GHz Centrino Duo with 1GB RAM. The video sequences were acquired at 24 frames per second. The proposed approach was implemented in Microsoft Visual C++. Figure 7 shows both the lab simulator and the in-vehicle configuration. While preliminary tests have shown that the proposed approach works well in the in-vehicle environment, all results reported here have been obtained using the lab simulator.

a)



b)

Figure 7. System set-up: a) lab simulator; b) in-vehicle configuration.

Figure 8 shows a snapshot of the graphical interface used for off-line calibration, as well as for on-line video processing and eye-state detection. As a result of real-time processing, the result of eye-state detection (open-eyes in Figure 8) is computed and displayed synchronously with the corresponding input frame from the webcam.

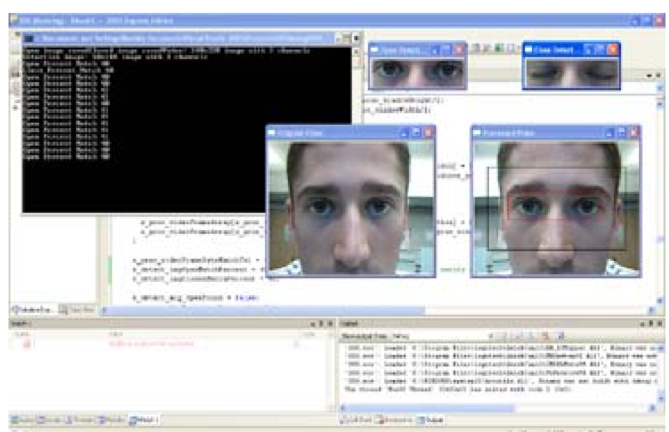


Figure 8. Snapshot of graphical interface used for the implementation and testing of the proposed approach

Our proposed approach models eye blinks as a two-state process, without an explicit consideration of transitory states from eyes-open to eyes-closed or vice versa. Thus, a first experiment measured the impact of transitory states on the overall performance of the eye-state detection approach. In order to isolate the transitory state phenomenon from other sources of error (head motion, variable distance between the ey-plane and the image plane etc.) the video sequences used in this experiment were acquired at a close distance from the webcam and without any perceivable head motion. A typical result is shown in Figure 9.

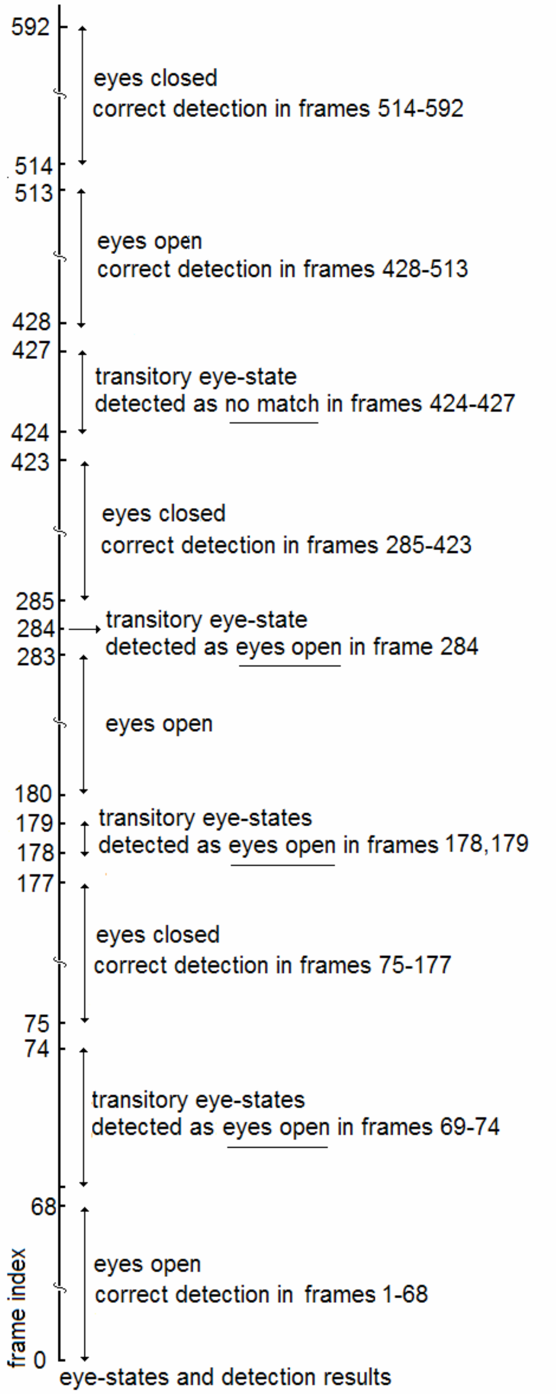


Figure 9. Eye-states and detection results in video sequence with stationary head

The results in Figure 9 show that transitory eye-states lying in-between eyes-open and eyes-closed induce a certain amount of error in the eye-state detection process. The correct detection result for all transitory states would be 'no match'; however, one may expect some transitory states to be more similar to eyes-closed, and others to be more similar to eyes-open. One may also notice from Figure 9 that the number of transitory states in-between eyes-open and eyes-closed is low, ranging from 0 to 5 frames. Therefore, given the fact that closed eyes need to be consistently detected for 3 seconds (i.e. in n=72 frames at 24 fps) in order to trigger the warning alarm, the errors introduced by transitory states in the sleep onset detection are negligible.

In video sequences containing a certain amount of head motion and allowing a variable distance from the subject to the image plane during calibration, the results of eye-state detection are less than 100%. Our experimental results were computed over a database of 30 sequences acquired with 15 different subjects and using the lab simulator shown in Fig. 5a. The subjects were asked to maintain eye-closed states for short periods of time (regular eye-blinks), as well as for longer periods of time (more than 3 seconds), in order to simulate sleep onset. The average length of video sequences was 600 frames, acquired at 24 frames per second. All sequences were processed using the default set of values for all parameters used by the proposed approach (i.e. the size of the search region for template matching, size of eye-state templates, thresholds τ_0 and $\tau_{R,}$ and n=72 frames of successive eyes-closed detections for alarm triggering).

The global performance of our algorithm for the video database is summarized in Table 1. The performance statistics in Table 1 were computed against a ground-truth labeling of eye-states, which was performed on a frame-by-frame basis by a human operator. One may notice from Table 1 that the rate of correct alarm triggers is higher than the rate of correct eyes-closed detections. This result is due to the fact that most of the missed detections for the eyes-closed state happen during regular eye-blinks. Eyes-closed states that were maintained over longer periods of time resulted in higher detection rates, as shown by the correct alarm rate.

Table 1. Performance statistics for proposed approach

No. of video sequences	30 sequences
Open Eye Detection	85%
Closed Eye Detection	70%
Successful Alarm Triggering	86%
Failed or False Alarm Triggering	14%

Figure 10 shows examples of eye-state detections obtained for three subjects in variable lighting conditions.

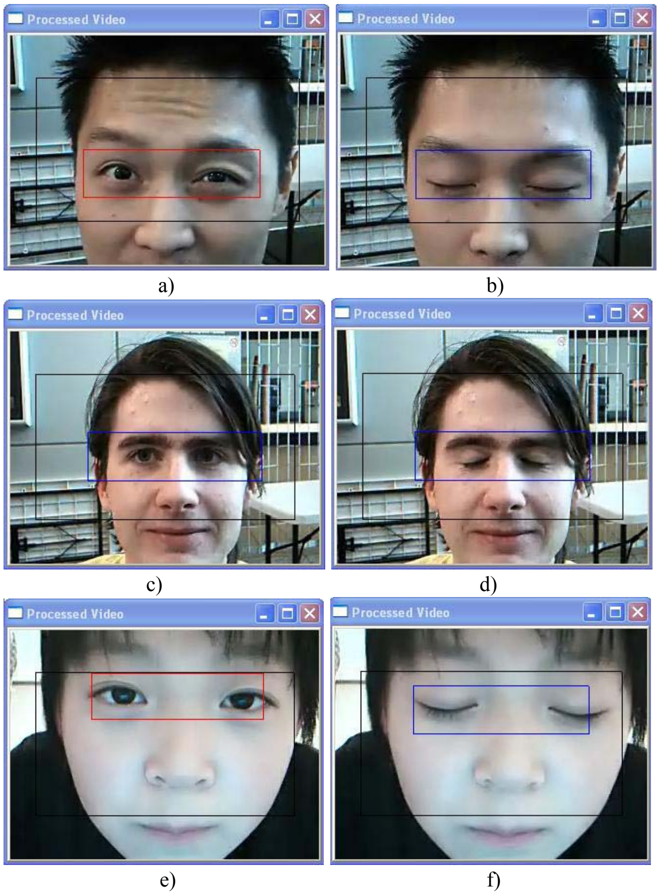


Figure 10. Results of eye-state detection for three different users in variable environmental conditions; Left column shows eyes-open states, while right column shows eyes-closed states. Red signals open-eyes detection, while blue signals closed-eyes detection.

The examples in Figure 10 show that our proposed approach works well with variable ambient illumination, skin color and eye shape. One source of error is the distance from the user's eyes to the image plane (see Fig. 8c); if this distance is too large, then the calibration box contains less pixels located in the eye regions. This increases the similarity between eyes-open and eyes-closed templates, and consequently the probability of confusing the states during template matching. Preliminary tests showed that, in case of an in-vehicle simulator, the distance from the webcam to the eyes plane is more constrained than for the current lab simulator. Specifically, this distance takes values within a range that allows for a good discrimination between eyes-open and eyes-closed templates.

IV. CONCLUSIONS

This paper proposes a novel approach for the real-time detection of sleep onset in fatigued drivers. Sleep onset is the most critical consequence of fatigued driving, as shown by statistics of sleep-related crashes. Therefore, unlike previous related work, we separate the issue of sleep onset from the global analysis of the physiological state of fatigue. This separation allows us for formulating our approach as a real-time event-detection problem.

Real-time performance was achieved by focusing on one single visual cue (i.e. eye-state), and by a custom-designed template-matching algorithm for on-line eye-state detection. The proposed approach also features an off-line calibration step, which provides subject-specific templates for the two eye-states of interest, and enables the user to adjust the parameters of the proposed approach.

Experimental results show that our approach works well in the current laboratory implementation. Ongoing work focuses on testing the proposed approach using the in-vehicle simulator. Future work will concentrate on both algorithmic and technological improvements to the current sleep surveillance system. From an algorithmic point of view, we are interested in increasing the robustness of the proposed approach to head motion; techniques for parametric and deformable template matching will be investigated for this purpose. From a technological standpoint, future work will focus on monitoring both day and night driving by using recent infrared webcam technology, whose performance and price are constantly improving. Once the proposed sleep surveillance system is successfully tested, the graphical interface that is currently implemented on a laptop device will be transferred on a dedicated dashboard display. Moreover, the proposed technique for computer vision-based sleep surveillance will be implemented via custom-designed DSPs or FPGAs embedded in the vehicle.

REFERENCES

[1] D. J. Beirness, H. M. Simpson, and K. Desmond "The Road Safety Monitor 2004. Drowsy Driving." Technical report. Ottawa, Ontario: Traffic Injury Research Foundation, 2005;

[2] J.A. Horne and L.A. Reyner, "Sleep-related vehicle accidents", British Medical Journal, vol. 310, pp. 565-567, 1995.

[3] A. Heitmann, R. Guttkuhn, A. Aguirre, U. Trutschel, and M. Moore-Ede, "Technologies for the monitoring and prevention of Driver Fatigue", In Proc.. of Int. Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, pp. 82-86, 2004.

[4] Q. Ji and X. Yang, "Real-time visual cues extraction for monitoring driver vigilance", in Proc. of Int. Symp. on Computer Vision Systems, pp. 107-124, 2001.

[5] Z. Zhu and Q. Ji, "Real-Time and non-intrusive driver fatigue monitoring", in Proc. of IEEE Intelligent Transportation Systems Conference, pp. 657-662, 2004.

[6] L. M. Bergasa, J. Nuevo, M. A. Sotelo, and Manuel Vazquez, "Real-time system for monitoring driver vigilance", IEEE. Int. Conf. on Intelligent Vehicles, pp. 78-83, 2004.

[7] P. Smith, M. Shah, and N.da Vitoria Lobo, "Monitoring head/eye motion for driver alertness with one camera", in Proc. of IEEE Int. Conf. on Pattern Recognition (ICPR), vol. 4, pp. 4636-4643, 2000.

[8] P. Smith, M. Shah, and N.da Vitoria Lobo, "Determining driver visual attention with one camera", IEEE Trans. on Intelligent Transportation systems, vol. 4:4, pp. 205-218, 2003.

[9] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical Pattern Recognition: A review", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22:1, pp. 4-37, 2000.