Real-time eye blink detection with GPU-based SIFT tracking

Marc Lalonde, David Byrns, Langis Gagnon, Normand Teasdale†, Denis Laurendeau‡

R&D Dept., CRIM, 550 Sherbrooke West, Suite 100, Montreal, QC, Canada, H3A 1B9
{marc.lalonde, david.byrns, langis.gagnon}@crim.ca
†Dept. of Social and Preventive Medicine (Kinesiology), Laval University,
Quebec City, QC, Canada, G1K 7P4
normand.teasdale@kin.msp.ulaval.ca
‡Dept. of Electrical and Computer Engineering, Laval University,
Quebec City, QC, Canada, G1K 7P4
denis.laurendeau@gel.ulaval.ca

Abstract

This paper reports on the implementation of a GPU-based, real-time eye blink detector on very low contrast images acquired under near-infrared illumination. This detector is part of a multi-sensor data acquisition and analysis system for driver performance assessment and training. Eye blinks are detected inside regions of interest that are aligned with the subject's eyes at initialization. Alignment is maintained through time by tracking SIFT feature points that are used to estimate the affine transformation between the initial face pose and the pose in subsequent frames. The GPU implementation of the SIFT feature point extraction algorithm ensures real-time processing. An eye blink detection rate of 97% is obtained on a video dataset of 33,000 frames showing 237 blinks from 22 subjects.

1. Introduction

The aim of this paper is to report about a first version of a real-time eye blink detector in the context of car driving simulation. This detector was designed within the COBVIS-D project [2], whose goal is to develop a simulation environment with a multi-sensor data acquisition and analysis system for driving performance assessment, cognitive load measure and training. Concretely, subjects are asked to sit in a driving simulator and react to realistic scenarios while being monitored by head and gaze trackers as well as monochrome video cameras. Their cognitive load will vary according to the degree of complexity of the driving task, and it can be assessed by analyzing and characterizing various physiological and physical data, among which are facial features.

The analysis of facial features can be carried out within the framework of the Facial Action Coding System (FACS) that allows the decomposition of facial expressions in terms of facial feature displacements called Action Units (AU) [5]. Literature review conducted for the COBVIS-D project has identified the FACS-based approach as the more appropriate to in-car driving situations. Workload or fatigue induces slight facial changes that cannot be detected with other methods. One can mention also that there are other physiological measures like electrocardiography and skin conductivity that might be better to characterize mental workload but these types of data were not available due to design constraints. According to the physiology literature, the main facial muscles that reflect mental effort are the *lateral frontalis*, the *corrugator supercilii*, *orbicularis oris* and *levator palpebrae superioris* ([13]). These muscles are responsible respectively for facial feature changes like eyebrow raiser, eyebrow frowning, lip suck, and eye blink. Eyebrow frowning is known to increase with concentration [12]. Eye blink rate, duration and amplitude are also known to vary with cognitive effort [16] and fatigue [11], [15].

This work focuses on the detection of eye blinks as a facial feature of choice for the measurement of the cognitive load. Detecting eye blinks implies finding and possibly tracking the eye region in order to reliably detect eyelid movements. In the literature, most of the works for eye detection and tracking are based on traditional optical passive acquisition set-ups (e.g. see [14] and references therein) with various algorithmic approches using appearance-based, model-based, feature-based or motion-based methods ([7]). Active acquisition set-ups have also been proposed based on near-infrared illumination that exploits the "red-eye effect" generated by a specially designed black and white camera with switched infrared lighting [6], [18]. There also exist a few practical facial recognition systems based on AU detection that capture eye blink information, notably from UCSD [3] and CMU [9].

The image acquisition set-up of our system is different and has been designed as a trade-off between many hardware and user specifications. The acquisition is done in the dark to maximize the illumination of the display showing the driving scenarios to the driver. The driver's face is illuminated with a near-infrared source and captured with a standard black and white CMOS camera (640x480 pixels). As a result, the images are of very low contrast, especially for the eye region. The nature of the images is an important aspect of the project and it justifies the design of a dedicated eye detection algorithm.

The paper is organized as follow. Sections 2 and 3 describes the eye and blink detection algorithms that have been implemented, and Section 4 gives performance results obtained so far.

2. Eye detection and tracking

Eye blink detection obviously implies prior detection of the eyes in the image of the subject's face. Tracking may also help stabilize the process since frame-based eye detection is not trivial, especially when considering the nature of the input images: low intensity combined with infrared illumination make eye pupils barely visible. One benefit from this is that the scene background is very dark, which greatly facilitates face location in the image.

2.1. Profiles for eye detection

A simple eye detector based on profile analysis [10] was considered. This method finds facial features by analyzing the horizontal profile (row grayscale averaging) of the face image. For example, the eyes 'create' deep valleys in the profile; the nose and the mouth have similar impacts. So locating the eyes amounts to finding the minimum of the large valley in the profile of the upper part of the face image. Once the row including the eyes is found, a vertical profile is computed and analyzed in order to find two deep valleys corresponding to the x-coordinates of the eyes. The nose is found in a similar way. As it is suggested in [10], the final position of the facial features may be determined after finding the best 'constellation' of features that corresponds to a human face. One major limitation is the high risk of failure as the subject rotates his head (in-plane). In addition, tests indicate that although fairly good precision is expected for the y-coordinate of the eyes, the valleys along the horizontal profile are quite spread with no real minimum, so x-coordinates are much less stable from frame to frame. In order to cope with these limitations, the profile-based technique is used as an initialization step for a feature tracking approach that is described next.

2.2. Why tracking

The idea behind resorting to tracking is this one: if one assumes that regions of interest can be found around the eyes at frame 0, accurate tracking of the head movement would allow adaptation of the position/shape of these regions. The key to adaptation is the ability to estimate the affine transformation between the face at current frame and the face at frame 0 (where it is assumed to be fully frontal). Yao et al. [17] opted for this approach to estimate the 3-D head pose based on two assumptions: facial features lie in a rigid plane, and change in position of the projected features is approximated by a global affine transformation. They used correlation to find correspondences between two consecutive frames and then estimated the parameters of the affine transformation, with temporal stability ensured by a Kalman filter. This paper reuses the same scheme but instead of computing correlations, which may be unreliable considering the poor quality of the images being processed, correspondences are found by comparing SIFT points in the current frame with those found in frame 0; regions of interest such as those including both eyes (denoted $R^0_{Left}$and R^0_{Right}) and defined at frame 0 will be subject to the same transformation that maps matching SIFT points.

2.3. Feature point extraction

Feature points are found using the scale-invariant feature transform (SIFT) [8]. They usually coincide with facial features such as nostrils, but stable points are also found on the head strap worn by the user during experiments. Although quite a few SIFT implementations are available as C/C++ code on the Internet, the GPU-based implementation provided in the library OpenVIDIA [1] was selected. It exploits the processing power of the graphics card to achieve a significant speedup (10x) over "traditional" software versions. Speeds around 60 frames per second (640x480 pixels in size) have been reached with an off-the-shelf nVIDIA graphics board that carried out both feature extraction and matching while at the same time relieving the main CPU.

2.4. Computing transformations

Once feature points $\mathbf{x_i'}$, are extracted from the current frame, the correspondence stage tries to match them to the points $\mathbf{x_i}$ extracted at frame 0 by comparing their signatures (Euclidean distance). Given N matching pairs ($\mathbf{x_i}$, $\mathbf{x_i'}$), with the assumption that

$$\mathbf{x_i'} = A\mathbf{x_i} + \mathbf{b} + \mathbf{e_i}, \qquad 1 \leq i \leq N \tag{1}$$

where $\mathbf{A}$ is the dilation/shear/rotation component and $\mathbf{b}$ is the translation component of the affine transformation ($\mathbf{e_i}$

is the model fitting error at each point), the least-squares estimate of the affine transformation is given by:

$$\hat{\mathbf{a}} = (X^T C^{-1} X)^{-1} X^T C^{-1} y$$

where $\hat{\mathbf{a}} = [\hat{A}_{11}, \hat{A}_{12}, \hat{b}_1, \hat{A}_{21}, \hat{A}_{22}, \hat{b}_2]$ represents the estimates of the affine parameters, $\mathbf{C}$=diag$(C_1, C_2, ...C_N)$ is a matrix of covariance estimates representing a measure of uncertainty about the matching 'quality' of each pair, and

$$X^T = \begin{bmatrix} x_1 & \mathbf{0} & ... & x_N & \mathbf{0} \\ 1 & 0 & ... & 1 & 0 \\ \mathbf{0} & x_1 & ... & \mathbf{0} & x_N \\ 0 & 1 & ... & 0 & 1 \end{bmatrix} \qquad y = \begin{bmatrix} x_1^{'} \\ x_2^{'} \\ ... \\ x_N^{'} \end{bmatrix}$$

and $\mathbf{0}^T = [0, 0]$. The matrix $\mathbf{C}$ could be built by using the distance between matching points that is computed during correspondence. Corners of the regions of interest $R^j_{Left}$and R^j_{Right} around the eyes at frame j are then given by (1) using corners of R^0_{Left} and R^0_{Right} as x_i.

2.5. Initialization

The profile method discussed in Section 2.1 may generate costly errors as time goes on but it is adequate for initializing tracking, provided that the user faces the camera at frame 0 (which is a reasonable constraint). Since the method gives approximate eye positions, especially for the x coordinate, adjustments are needed: the ROI presumably including the left eye is moved around its initial position and for each of those positions, the ROI including the right eye is also moved around its starting position. All visited pairs of ROI locations are characterized using three quality measures (between 0 and 1) that are added together to form a global score that should be minimized. The first measure is based on the difference of proportions of dark pixels between the center part of the ROI (analysis windows shown in light blue in Figure 1) and a region below (violet boxes). This ensures that each ROI is centered on the pupil and not the eyebrow. The second measure checks for the horizontal symmetry of each eye region. Finally, the last measure is a correlation coefficient that compares both regions (ROIs well aligned on eyes should have high similarity). The pair of ROI locations with the highest score represent R^0_{Left} and R^0_{Right}.

3. Blink detection

The basic step in blink detection is a simple motion detector based on thresholded frame difference inside the tracked regions of interest R^j_{Left} and R^j_{Right}. Optical flow is then computed inside the bounding box of each resulting blob to determine eyelid raise or fall. Although based on the algorithm in Bashkar et al. ([4]), which used motion flow to detect downward moving eyelids and initialize tracking of the eyes, the proposed approach rather relies on the tracked eye's position for local computation of the motion flow. The basic algorithm is described in Algorithm 1.

Algorithm 1 Algorithm for blink detection. See Figure 2.

1. *Locate motion regions using frame differentiation in each eye roi.*
2. *Threshold the motion regions and keep the better blob based on position, area, angle, density and width/height ratio.*
3. *Repeat 1-2 until candidates in both regions are found.*
4. *Compute optical flow field in the blob regions and extract the vertical and horizontal vectors in the blobs' coordinate system.*
5. *If the dominant motion is downward for both regions, the closing frame is saved otherwise steps (1) to (5) are repeated.*
6. *Repeat steps (1) to (4) with the additional constraint which is that blob position must be close enough to the saved closing position.*
7. *If an opening frame is found within the determined maximum blink length, the blink sequence is kept for further analysis, otherwise the blink hypothesis is rejected and the process restarts from step (1).*

In the process, blob filtering is obviously required in order to decrease false alarms. Not surprisingly, it appeared during development that a fixed set of thresholds on blob properties (area, angle, etc.) could not be set optimally for all subjects. So a set of adaptive thresholds that evolve over time has been created to match more accurately the blob properties associated to a given subject. When a new blink is detected, the model is updated by tightening thresholds, and tolerance to false alarms decreases. However, despite model update, some false alarms persist. After a completed blink, a validation step is performed : the ROI at the beginning of the blink is compared (square difference) to those throughout the blink sequence, thus making up a signature with properties such as amplitude and duration that are typical for each subject. A blink sequence should then meet the following three factors to be considered valid.

1. A local maximum in the signature must be present.
2. The local maximum must have non-negligible amplitude.
3. Local maxima from both eyes must be similar.

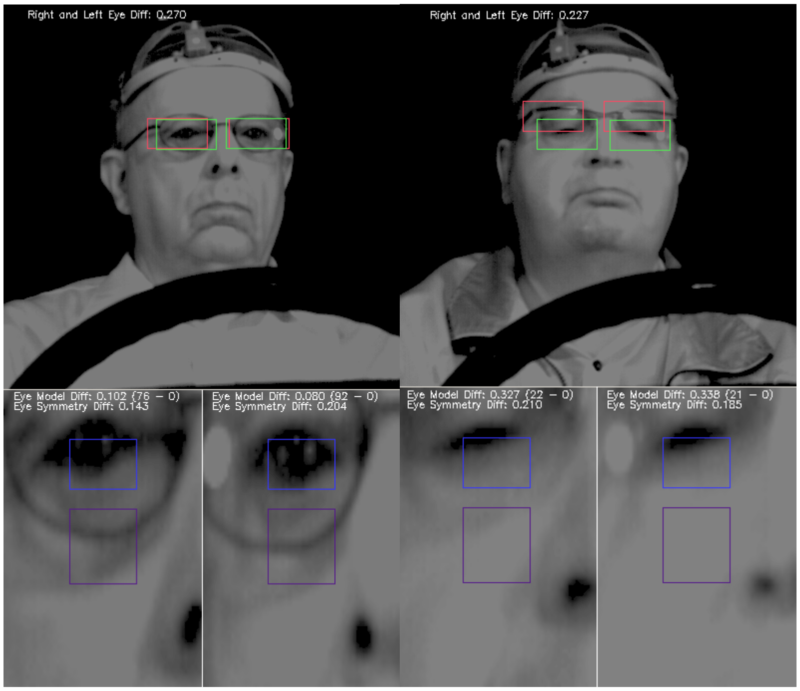

Right and Left Eye Diff: 0.270
Right and Left Eye Diff: 0.227
Eye Model Diff: 0.102 (76 – 0)
Eye Symmetry Diff: 0.143
Eye Model Diff: 0.080 (92 – 0)
Eye Symmetry Diff: 0.204
Eye Model Diff: 0.327 (22 – 0)
Eye Symmetry Diff: 0.210
Eye Model Diff: 0.338 (21 – 0)
Eye Symmetry Diff: 0.185

Figure 1. In the upper images, red boxes show the approximate eyes position based on image profiles and the green ones are the best positions found. The lower images show left and right eye regions with the desired eye location in light blue. The ROI will be well aligned with the pupil when the number of dark pixels in the light blue box/violet box are maximal/minimal. Note that the images shown here, as well as in the remainder of the paper, have been enhanced (histogram-equalized) for clarity.

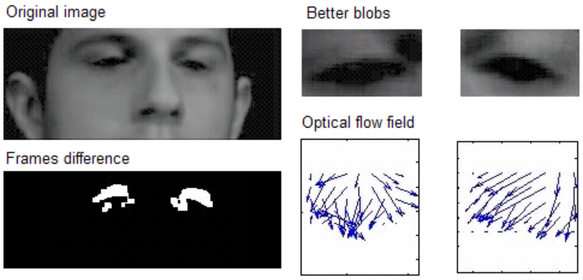


Figure 2. Step 1 of the blink detection algorithm is illustrated at the bottom left showing blobs from frame difference. Upper right: ROIs corresponding to the better blobs of step 2 are shown. Bottom right: the optical flow field computed for each blob is shown, and a dominant downward motion can easily be determined.

Figure 3 shows various types of false alarms, some that don't reach the minimum square difference, and some without local maxima.

4. Results and discussion

The dataset used for testing the algorithm is composed of raw (uncompressed) video sequences of 22 subjects sitting in the driving simulator (one sequence per subject, with an average length of 45 seconds, or 1,500 frames). Among the subjects, there are nine elderly persons, and eight participants wearing glasses.

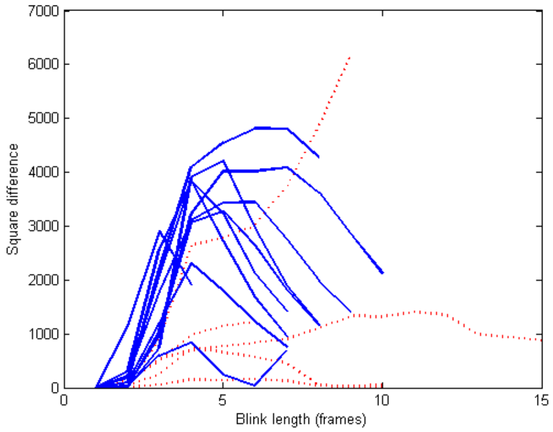


Figure 3. This graph shows the square difference between first and remaining images of a detected eye blink sequence, for many sequences. Solid (blue) lines show valid blink sequences and dotted (red) lines show some false alarms. One source of false alarm is a brief head movement, whose amplitude generates different signatures.

SIFT feature point extraction yields around 120 matching points in the face area. (Recall that matching is performed between the set of points extracted at frame 0 and those at the current frame.) Roughly 80 matching pairs are

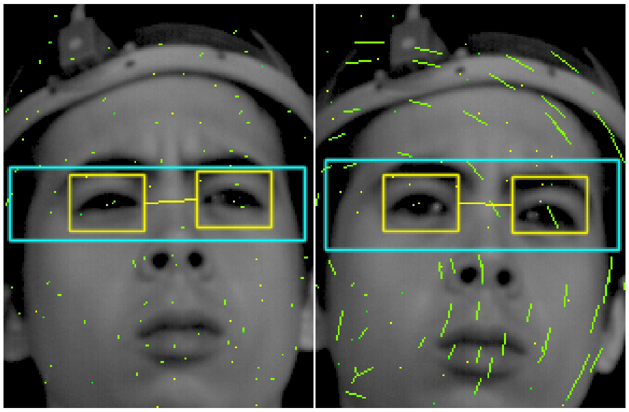


Figure 4. Set of correspondences found when SIFT points from rotated face at frame n (right) are matched to reference face at frame 0. Green lines link matching points.

coherent enough to be used for the estimation of the affine parameters. This number decreases to 50-60 when in-plane face rotations occur (see Figure 4 for an example), and naturally drops significantly when out-of-plane rotations approach 90 degrees (subject wants to look to the side). In this latter case, loss of track eventually occurs due to the lack of reliable correspondence between the frontal and profile views. One interesting benefit of the method is that as the head turns back into normal position, an increasing number of matching points are found and tracking normally resumes without external intervention (Figure 5).

It could be argued that SIFT feature point matching may not yield a reliable estimate of the affine transformation because of the non-rigid deformations of the face due to face expressions. Typical experimental conditions are such that subjects perform the required driving tasks with a neutral face facing the camera. Loss of track and blink misses as a result of marked head turns are not critical for the experiments. Furthermore, significant face areas (head strap, nose, ears) can be considered rigid and are thus expected to supply reliable points. A potential improvement would be to evaluate the reliability of the feature points and use the reliability measures during the estimation of the affine parameters.

As far as blink detection per se is concerned, ground truth analysis shows that each sequence contains from 1 to 48 blinks for a total of 237 blinks over all sequences. A blink detection rate of 97% was obtained (only 7 missed blinks), whereas 25 false detections were found over the >33,000 frames processed. The algorithm was tuned to be very sensitive to the vertical movements in order to account for some subjects having eye blinks of small amplitude (Figure 6). Closer analysis of these 25 false detections reveals that they were the result of gaze lowering (16), vertical head movements (5), camera vibrations (3) and eye movements (1). Gaze lowering regularly occurs during the driving simulation. Although a decrease in the number of false detections is desirable, the current performance is reasonably good considering the low image quality. In the current implementation, crude filtering of the parameters of the affine transformation ensures tracking smoothness; a more formal filter design based on Kalman filter is under way, but it is not clear whether a better filtering will reduce false detections significantly since blob filtering during blink detection is already tolerant to light tracking jitter. In the same vein, the sensitivity of the tracking to facial expression changes should be evaluated as well as their impact on blink detection. As for processing speed, the current version with non-optimized code (apart from the GPU-based SIFT library) reaches about 25 frames per second. A screenshot of the module is shown in Figure 7.

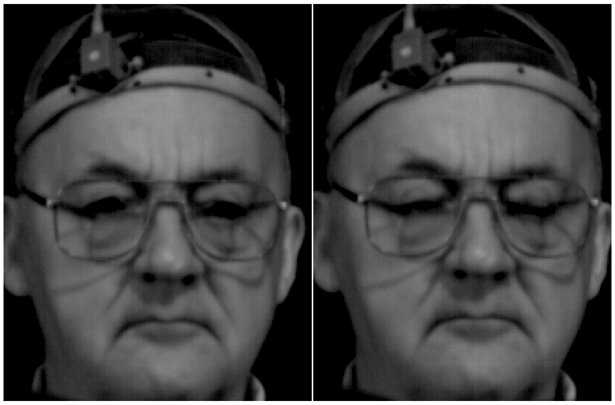


Figure 6. Difficult case in blink detection; nevertheless the algorithm successfully detected the blink.

5. Conclusion

In conclusion, this paper presented a blink detection algorithm, combined to SIFT-based tracking for greater robustness. Tracking was performed in near real time thanks to the use of a GPU-based SIFT implementation. Detection rate was 97% over 22 sequences averaging 1,500 frames in length, and false alarms were reasonably low (less than 1 in 1,000 frames). The tracking capability developed in the context of blink detection is a key feature for future work: the stable reference frame that it provides will be a sound basis for a more extensive analysis of facial features related to drivers' cognitive load such as frowning: one merely needs to define and track more regions of interest over the eyes for e.g. capturing eyebrow movements.

Acknowledgements

This work is supported in part by (1) Société de l'Assurance Automobile du Gouvernement du Québec (SAAQ), (2) AUTO21 research network, (3) Natural Science and Engineering Research Council (NSERC) of

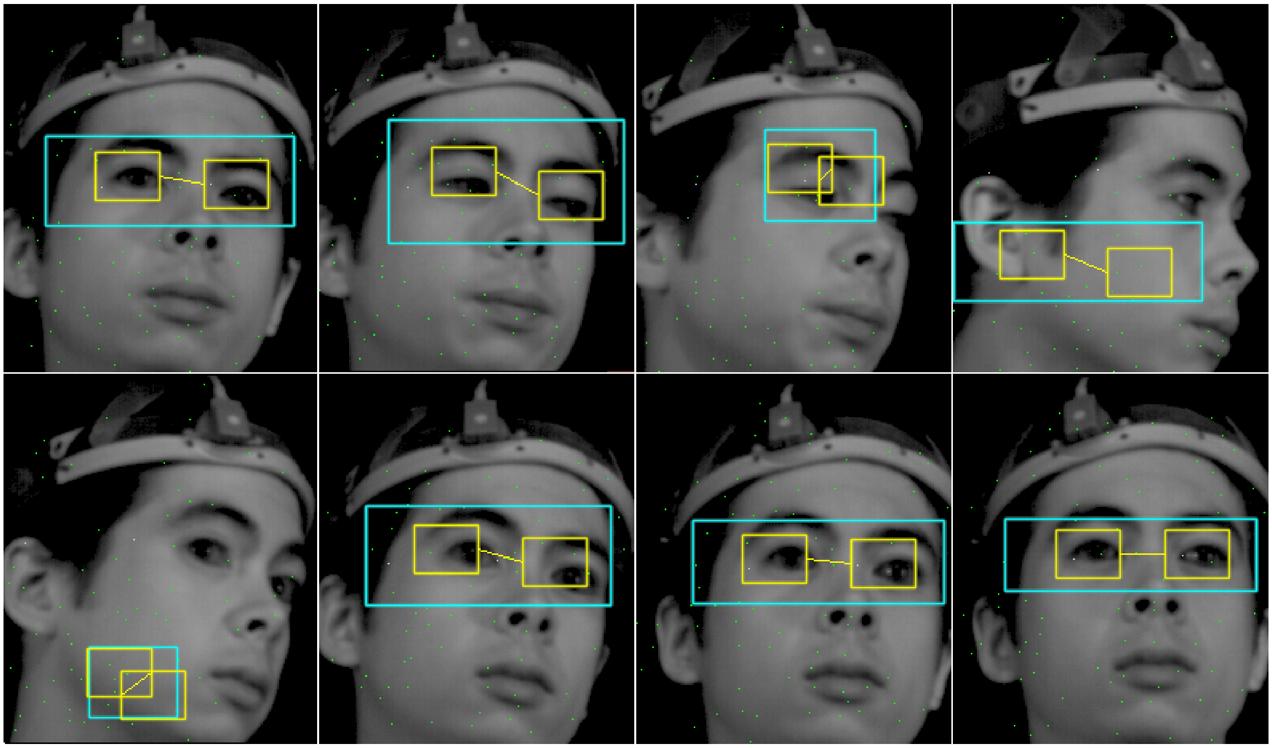


Figure 5. Sub-sampled sequence with subject looking to the side. Tracking eventually resumes automatically.

Canada and (4) Ministère du Développement économique de l'Innovation et de l'Exportation (MDEIE) du Gouvernement du Québec. We also thank Dr. France Laliberté (University of Montreal Hospital Research Center) who participated in the early phase of this work.

References

[1] OpenVIDIA : Parallel GPU Computer Vision. `http://openvidia.sourceforge.net`. Accessed December 11, 2006.

[2] Auto21. The automobile of the 21st century. `http://www.auto21.ca/intelligentsystems4_e.html`. Accessed Mar. 13, 2007.

[3] M. Bartlett, G. Littlewort, B. Braathen, T. Sejnowski, and J. Movellan. A prototype for automatic recognition of spontaneous facial actions. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 1271–1278, 2003.

[4] T. N. Bhaskar, F. T. Keat, S. Ranganath, and Y. V. Venkatesh. Blink detection and eye tracking for eye localization. In *Proc. TENCON 2003*, volume 2, pages 821– 824, 2003.

[5] P. Ekman. *Facial Action Coding System: a technique for the measurement of facial movement*. Consult. Psych. Press, 1978.

[6] A. Haro, M. Flickner, and I. Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 163–168, 2000.

[7] Q. Ji, H. Wechsler, A. Duchowski, and M. Flickner. Editorial: special issue: eye detection and tracking. *Comput. Vis. Image Underst.*, 98(1):1–3, 2005.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. of Computer Vision*, 60(2):91–110, 2004.

[9] T. Moriyama, T. Kanade, J. Cohn, J. Xiao, Z. Ambadar, and H. Gao. Automatic recognition of eye blinking in spontaneously occurring behavior. In *Proc. Int. Conf. Patt. Rec.*, 2002.

[10] K. Sobottka and I. Pitas. A fully automatic approach to facial feature detection and tracking. In *Proc. Audio- and Video-Based Biometric Person Authentication*, pages 77–84, 1997.

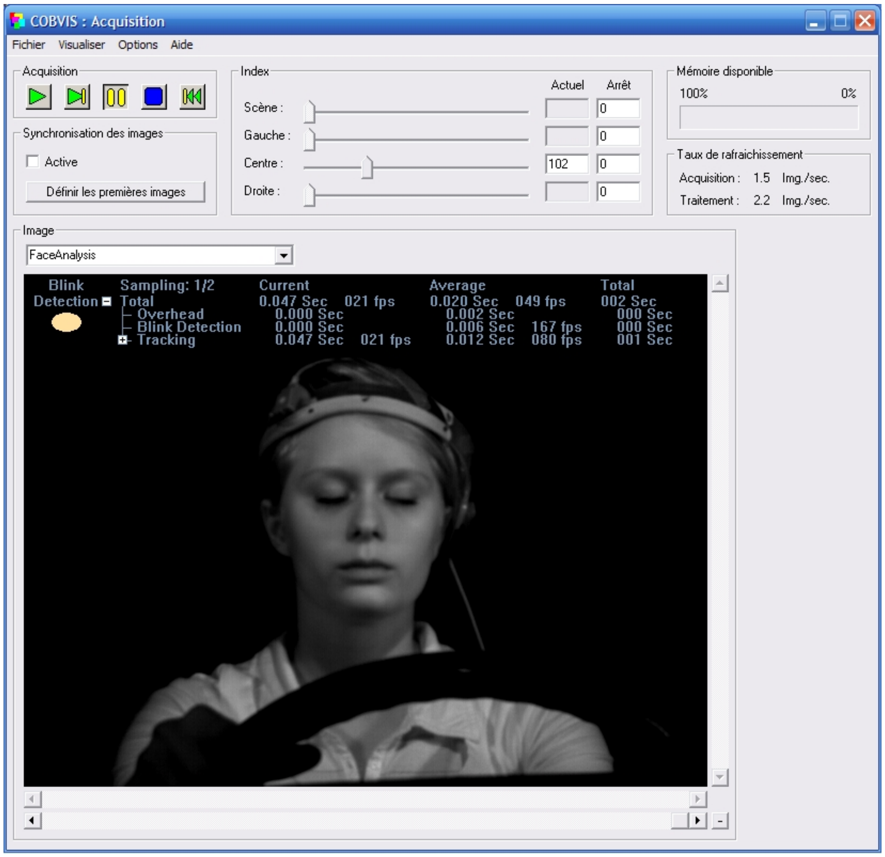
[11] J. A. Stern, D. Boyer, and D. Schroeder. Blink rate: a possible measure of fatigue. *Human Factors*, 36(2):285–297, 1994.

[12] M. Valstar, M. Pantic, Z. Ambadar, and J. Cohn. Spontaneous vs. posed facial behavior: Automatic analysis of brow actions. In *Int. Conf. on Multimodal Interfaces*, pages 162–170, 2006.

[13] D. D. Waard. *The measurement of drivers' mental workload*. PhD thesis, University of Groningen,Traffic Research Centre, 1996.

[14] P. Wang, M. B. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *IEEE Workshop on Face Recognition Grand Challenge Experiments (with CVPR)*, 2005.

[15] A. Williamson and T. Chamberlain. Review of on-road driver fatigue monitoring devices. Technical report, NSW In-



☐ Active

Index	Actuel	Arrêt
Scène :		0
Gauche :		0
Centre :	102	0
Droite :		0

Mémoire disponible: 100% 0%

Taux de rafraichissement
Acquisition : 1.5 Img./sec.
Traitement : 2.2 Img./sec.

Image: FaceAnalysis

```
  Blink     Sampling: 1/2     Current                Average                Total
Detection ⊟ Total             0.047 Sec   021 fps    0.020 Sec   049 fps    002 Sec
            ├ Overhead          0.000 Sec              0.002 Sec              000 Sec
            ├ Blink Detection   0.000 Sec              0.006 Sec   167 fps    000 Sec
            ⊞ Tracking          0.047 Sec   021 fps    0.012 Sec   080 fps    001 Sec
```

Figure 7. Output screen of the eye blink detection module.

jury Risk Management Research Centre, University of New South Wales, April 2005.

[16] G. Wilson. An analysis of mental workload in pilots during flight using multiple psychophysiological measures. *Int. J. of Aviation Psychology*, 12(1):3–18, 2002.

[17] P. Yao, G. Evans, and A. Calway. Using affine correspondence to estimate 3-d facial pose. In *Proc. 8th Int. Conf. on Image Processing (ICIP 2001)*, pages 919–922, 2001.

[18] Z. Zhu, Q. Ji, and K. Fujimura. Combining kalman filtering and mean shift for real time eye tracking under active ir illumination. In *Proc. International Conference on Pattern Recognition*, pages 318–321, 2002.