A computer vision system for analyzing and interpreting the cephalo-ocular behavior of drivers in a simulated driving context

S. Metari[1], F. Prel[1], T. Moszkowicz[1]
D. Laurendeau[1] and N. Teasdale[2]
[1]LVSN and [2]GRAME, Université Laval
Québec, Qc, Canada, G1V 0A6
{samy.metari.1, florent.prel.1}@ulaval.ca
{Thierry.Moszkowicz, Denis.Laurendeau}@ulaval.ca
Normand.Teasdale@kin.msp.ulaval.ca

S. Beauchemin
Department of Computer Science
The University of Western Ontario
London, ON, Canada, N6A 5B7
beau@csd.uwo.ca

Abstract

In this paper we introduce a new computer vision framework for the analysis and interpretation of the cephalo-ocular behavior of drivers. We start by detecting the most important facial features, namely the nose tip and the eyes. For that, we introduce a new algorithm for eyes detection and we call upon the cascade of boosted classifiers technique based on Haar-like features for detecting the nose tip. Once those facial features are well identified, we apply the pyramidal Lucas-Kanade method for tracking purposes. Events resulting from those two approaches are combined in order to identify, analyze and interpret the cephalo-ocular behavior of drivers. Experimental results confirm both the robustness and the effectiveness of the proposed framework.

1. Introduction

Driving is a very important activity for a large portion of the population, especially among the elderly. Epidemiological studies show that this category of drivers may sometime adopt an unsafe driving behavior. This is due to the difficulties experienced by those drivers in demanding driving situations such as car overtaking, lane change, intersection crossing, etc. Those driving contexts involve a complex cephalo-ocular behavior and visual research actions, such as blind spot checking and rear view / lateral mirror verification. Evaluation and improvement of the driver performance in a safe environment (driving simulator) are of great importance for road safety. The main objective of this work is the elaboration of a new computer vision system for evaluating and improving driving skills of older drivers. This system imply the analysis and treatment of cephalo-ocular behavior and visual search actions of older drivers in a simulated driving context (cf. Figure 14). More specifically, we focus on the visual research action related to the verification of the blind spots when overtaking and lane changing. Experiments run in our laboratory have shown that 80% of older drivers do not check the blind spots in these contexts. Thus, by providing a new system allowing the objective detection of driving errors in a safe environment (simulator), it is expected that retraining drivers will lead to safer driving in the long term. The proposed system includes three main processing steps. The first one is devoted to the detection of the most important facial features, namely the nose tip and the eyes. The second one is dedicated to the tracking of these facial features. The last step deals with the analysis and interpretation of the cephalo-ocular behavior and the visual research actions of driver.

In section 2 we describe different approaches for the detection of facial features. Section 3 is devoted to the tracking process. The proposed system is detailed in section 4. Experimental results of intermediate steps of the system are shown in the appropriate subsection. Note that images and videos used in this paper can be both in color and in gray-level mode.

2. Detection of Facial Features

In pattern recognition, an important research field is the detection and localization of objects and patterns. The majority of existing research work [10, 11, 14, 20, 21] is based on the following approach: a sliding window is matched with image parts at different positions and scales. Each mapping reveals whether the sliding window contains the requested object or the background. Another use of this approach is to detect parts of the object instead of the whole object [8, 9]. Those detected parts will be assem-

978-0-7695-4040-5/10 $26.00 © 2010 IEEE
DOI 10.1109/CRV.2010.35

bled in order to recognize the full target. Another set of approaches [2, 4] is based on region mapping around extracted local interest points from the image, rather than performing operations on the whole image.

2.1. Eyes detection

In what follows, we introduce a new method for eye detection. The main goal is to identify the two pupils of a person. The proposed method is based on a priori knowledge of eye geometry, on its position in the face and on its position relative to the other eye (angle, distance, shape, etc). Additionally, the Region of Interest ROI of the eyes in the face is identified a priori. The recognition of the two pupils reduces to the identification of a pair of blobs (connected set of pixels) with relatively round shapes and reasonable sizes. The proposed method is comprised of the following steps:

Extraction of the blobs: Blob extraction as shown in Figure 4.(c) is achieved according to the two following steps: First, a saturation process is applied to the eyes ROI followed by the application of a closing operator (dilatation and erosion). Second, an algorithm for connected components extraction is applied to obtain blobs. The resulting blobs are characterized by different sizes and shapes and only two of them correspond to the eyes. In what follows, a serie of tests is applied in order to identify the requested pair of blobs.

Subdivision of highly non-convex blobs: The saturation process can generate highly non-convex blobs that may contain the requested blobs. For example, if we apply the saturation process to an image of a driver wearing eyeglasses, the eye can merge with a part of the glasses (cf. Figure 1). The shape of the blob containing the eye is thus highly non-convex and differs from the requested shape (eye geometry). One possible solution to this problematic situation is to subdivide the highly non-convex blob (cf. Figure 2.(a)) in several other convex blobs (cf. Figure 2.(b)). To this end, we introduce the following algorithm:

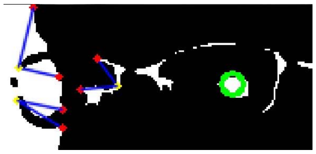
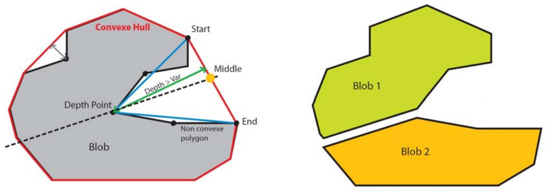


Figure 1. Example of non-convex blob.

- Calculation of the convex hull encompassing the considered blob using the algorithm proposed by Sklansky [17].
- Calculation of the difference between the convex hull of the blob and the blob itself. The OpenCV algorithm "CvConvexityDefect" used here returns the starting and end points of the non-convexity, the coordinates of the deepest point and the depth of each non-convexity.
- Filtering: we only consider the strong non-convexities, i.e. a depth of the order of several pixels. Thus, for non-convex regions whose depth is greater than n pixels, the blob is separated in two parts.
- Calculation of the line separating the blob: the latter connects the deepest point with the midpoint between the extremity points (cf. Figure 2.(a)).
- Subdivision of the blob: the elements of the blobs on either side of the line are grouped into two new blobs (cf. Figure 2.(b)).



(a) (b)

Figure 2. The principle of blob separation.

Selection of blobs: Once the blobs are detected and divided whenever necessary, a first filter is applied in order to eliminate those blobs whose features are not compatible with the typical shape of an eye. The selection criteria are very simplistic, but are discriminating enough for enabling the identification of the requested blobs:

- **The number of pixels:** must be within an acceptable range. Thus, a blob made up of less than 5 pixels is considered too small, but a blob of more than 200 pixels is too big (cf. Figure 4.(d)).
- **Dimensions:** The width and height of the rectangle enclosing the blob are calculated (cf. Figure 4.(e)). The ratio width/height can reveal the shape of the blob. Each blob elongated in the vertical direction is eliminated (width/height « 1).
- **The shape:** a circle C_{fit} is adjusted by a least squares method [17] for each of the blobs having passed the previous tests (cf. Figure 4.(f)).

  A first discrimination of the blobs is made according to the value of the radius of the circle.

A second discrimination is based on the ratio R_A between the surface covering the intersection between the blob and the circle and the surface of the circle itself. This ratio reveals whether the blob is circular and is well registered in the best circle C_{fit} (cf. Figure 4.(g)).

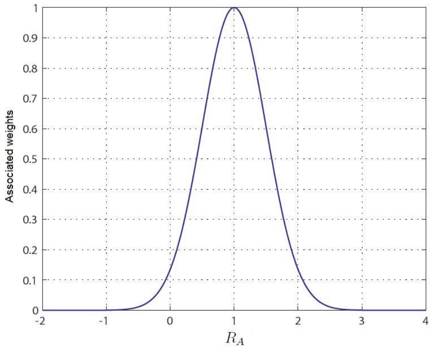
$$R_A = \frac{A_{blob} \bigcap A_{circle}}{A_{circle}}, \tag{1}$$

with A_{blob} and A_{circle} are respectively the area of the blob and the area of the circle in pixels.

The blobs that survived to the above mentioned criteria, are rated on a scale ranging from 0 to 1. To this end, a weighting system based on the Standard Gaussian SG distribution is used in order to associate a weight to each blob, according to its characteristics that are compared to an ideal case. A weight of 1 corresponds to a blob corresponding to the ideal case.

$$SG(x) = exp(-\frac{(x-\mu)^2}{2\sigma^2}), \tag{2}$$

with μ the mean and σ the standard deviation.

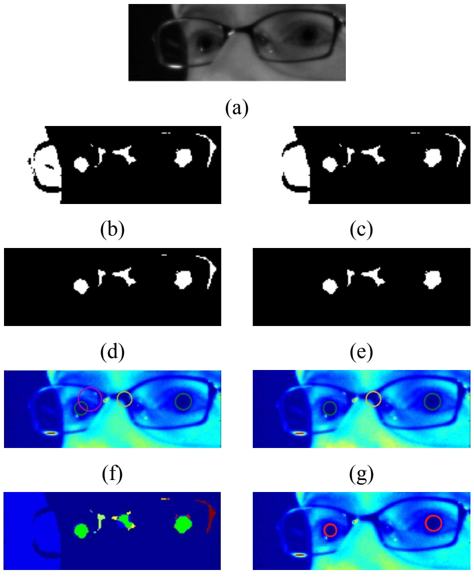


(a)

Figure 3. Standard Gaussian distribution with $\mu = 1$ and $\sigma = 0.5$.

The parameter μ represents the reference value, i.e. the value corresponding to an ideal ratio. It would be 1 if we consider that the pupil is perfectly circular regardless of the viewpoints. In this case, the used Gaussian is centered around 1 (cf. Figure 3). The parameter σ is calculated empirically and corresponds to the tolerance threshold given to the ratio values.

Finally, each blob has a weight associated to the ratio R_A, to the radius of the circle C_{fit} and to the ratio width/height. The final weight of the blob is given by the average value of these three weights.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 4. Steps of the eyes detection algorithm. (a): the region of interest, (b): after saturation of the image, (c): after closure, (d): after selection by size of blobs,(e): after selection by shape of blobs, f): fitting circles on blobs, (g): fit application on the image of ROI and selection according to the radius of fit the circle, (h): comparison between the surfaces of circles and the intersection of circle and blob, (i): final selection.

Selection per pairs of blobs: At this step of the process, we selected the blobs that have a high probability of corresponding to the eyes. In what follows, we do not consider blobs individually, but as pairs. We identify the pairs of blobs that are arranged such as to correspond to both eyes. In our working context (Driving simulator), the distance between the driver and the camera is roughly constant over time. Additionally, since the human morphology is relatively constant, we assume that the gap between the eyes varies slightly from one individual to another. These two simplifications are exploited thereafter and make it possible to effectively discriminate the blobs that can correspond to pairs of eyes.

We start by constructing, without repetition, all possible pairs of blobs. Knowing the number of blobs n, the number of possible pairs N is given by:

$$N = C_n^2 = \frac{n!}{2!(n-2!)}. \tag{3}$$

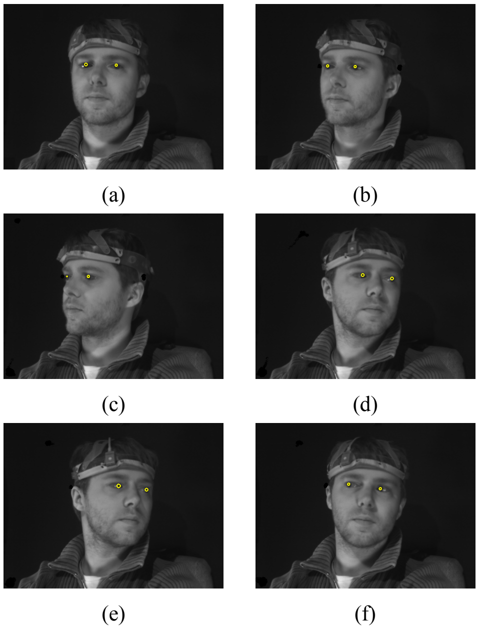
Once all possible pairs are built, the next step consists in calculating the Euclidean distance d_{pair} between the blob centers of each pair, as well as the angle α_{pair} between the line connecting the two blobs of the considered pair and the horizontal of the image. These criteria allows us to discriminate more pairs of blobs.

$$\begin{aligned} d_{pair} &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \\ \alpha_{pair} &= \arctan(\frac{y_2 - y_1}{x_2 - x_1}), \end{aligned} \tag{4}$$

with (x_1, y_1) and (x_2, y_2) the coordinates of the blobs centers.

For the remaining pairs, a weighting system similar to the one used for rating the blobs, is applied in order to compare the probability that a pair corresponds to the eyes. This system combines the information about both the pairs and the blobs. Thus, if two pairs have similar values of d_{pair} and α_{pair}, the values characterizing the blobs will make the difference. If by this process, the weights assigned to each pair do not allow the selection of the requested pair of blobs, the final selection will be achieved at the matching step.

Experimental results of the above mentioned approach are given in Figure 5. We conclude that the detection of both eyes is performed successfully and independently of the head position in the image.



(a) (b)

(c) (d)

(e) (f)

Figure 5. Experimental results of eyes detection.

2.2. Nose detection

For nose detection we call upon the cascade of boosted classifiers based on Haar like features technique. This machine learning approach for rapid object detection, was first introduced by Viola and Jones [19] and then extended by Lienhart and Maydt [6] using a new set of rotated Haar-like features. It is achieved according to the following steps. First an AdaBoost-based classifier is trained from a set of positive and negative examples. Positive examples are target images and negative examples are arbitrary images not including the target. Once the classifier is trained, the next step consists in target detection. For that, a sliding window is applied at different positions within the requested image. For each position, the classifier decides whether the sliding window contains the target or not. Finally, the method returns the regions likely to contain the target. Further details about the classifiers-based detection can be found in [6, 10, 19, 20].

In our case, for the training step we take a set of 500 images from which we construct a set of 2500 positive and 2500 negative examples (cf. Figure 6 and 7). Positive examples are images of different noses extracted manually from our image database. Negative examples are images of different parts of the face not including the nose. Experimental results are given in Figure 8.

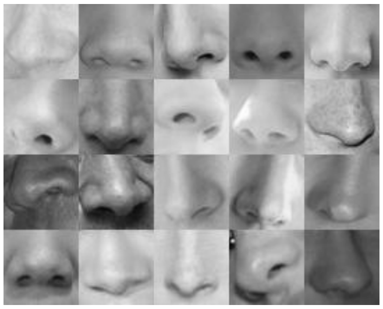


Figure 6. A sample of positive examples.

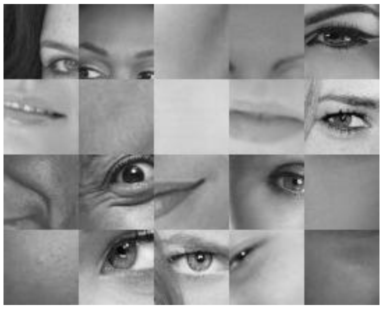


Figure 7. A sample of negative examples.

In figure 8, obtained results reveal that the detection of the nose tip is achieved successfully regardless of face orientation and skin color.

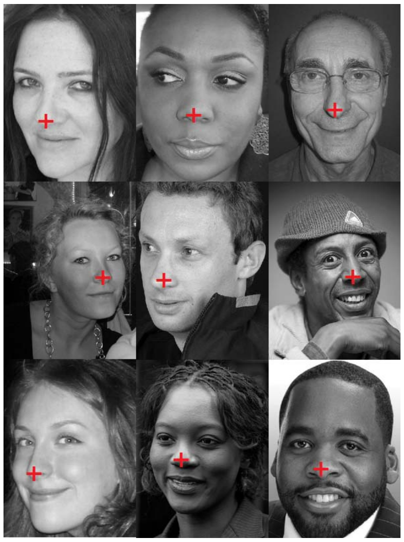


Figure 8. Nose tip detection results.

Note that the approach used for nose tip detection can be adapted to eyes detection as well. This is achieved by the use of a new learning set based on eye models. Note that we use the learning set provided by the OpenCV library. In Figure 9 we show the experimental results obtained with the joint detection of both eyes and nose tip.

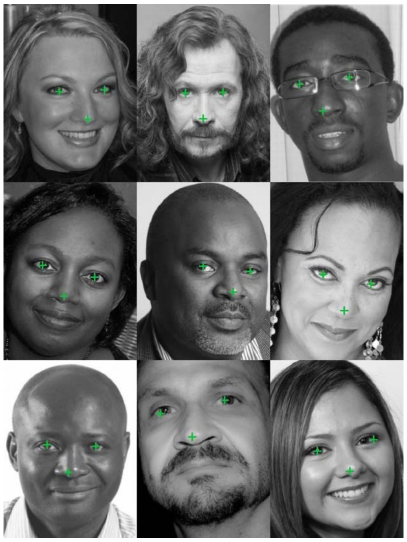


Figure 9. Joint detection of both eyes and nose tip.

3. Tracking of facial features

Once the facial features (nose tip and eyes) are well detected, the next step consists in tracking the features in video sequences. Object tracking is an important research field in the domain of computer vision. In the literature, we find three main families of approaches for object tracking. The first one is point-based tracking [12, 16, 18]: at each frame, the requested objects are detected and represented by points. The correspondence of points is achieved according to the previous state (position and motion) of the object. The second one is kernel-based tracking [3, 7, 15]: the requested object is modeled as a geometric template (triangle, rectangle, ellipse, etc). Object tracking is achieved by calculating the kernel motion across the frames. This motion is often modeled as a parametric transformation such as affine or similarity transformation (translation, scale, rotation). The latter is silhouette-based tracking [1, 5, 13]: tracking process is achieved by estimating the object region across the frames. The information encoded inside the object region are used in order to model the object. The tracking is performed by matching the silhouettes and object models.

Following an overview of the literature, we opted for the Lucas-Kanade (LK) method [7]. More specifically we used the pyramidal implementation of this method [2]. A summary of the problem statement of the LK method is described in the following.

Brief summary of the LK method: The LK algorithm is a two-frame differential method for optical-flow based motion estimation. This method is based on the assumption that the optical flow is constant at the local neighborhood of the considered pixel. Let us consider two gray-level images I and J of size $n_x \times n_y$. Taking a specific pixel $\vec{u}(u_x, u_y)$ from the first image I. The main goal of feature tracking is to find the location $\vec{v} = \vec{u} + \vec{d}$, on the second image J , such that $I(\vec{u}) \simeq J(\vec{v})$. The vector $\vec{d}$ corresponds to the image displacement. It is estimated by minimizing the residual function $\varepsilon(\vec{d})$, which is defined as follows [2]:

$$\begin{aligned}\varepsilon(\vec{d}) &= \varepsilon(d_x, d_y) = \sum_{x=u_x-\omega_x}^{u_x+\omega_x} \sum_{y=u_y-\omega_y}^{u_y+\omega_y} [I(x,y) \\ &- J(x+d_x, y+d_y)]^2. \qquad (5)\end{aligned}$$

Note that the local neighborhood is of size $(2\omega_x + 1) \times (2\omega_y + 1)$. Typical values for ω_x and ω_y are 1, 2, 3, ... pixels.

Between the well-known classical techniques, the least squares method is the most used for the resolution of the above system. However, the pyramidal implementation [2] of the classical Lucas-Kanade algorithm remains the most powerful solution to this problem.

In Figure 10 we show experimental results of nose tip tracking by selecting six different frames from the considered video. Note that the facial feature is well tracked throughout the video sequences.

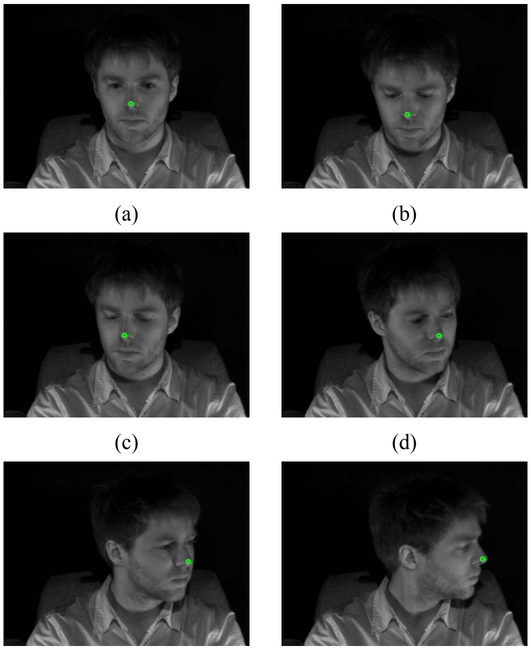


Figure 10. Nose tip tracking.

4. Analysis and interpretation of the cephalo-ocular behavior

The main goal of our work is to study the cephalo-ocular behavior of drivers. More specifically, we are interested in the study of the visual research actions related to the verification of the blind spot when changing lanes and overtaking. Recall that the blind spot is the space on each side of a car that is not covered by the driver's fields of vision (including the fields of vision due to rear-view and wing mirrors). In order to test if the driver is checking the blind spots, we propose the following algorithm:

1. Detection of the facial features (nose tip and eyes) using the techniques described in subsections 2.1 and 2.2.

2. Tracking of the facial features using the method described in section 3. The tracking process is accompanied by the calculation of different distances (cf. Figure 11.(b) and (c)) separating the facial features. Those distances allow us to identify the orientation of the driver's head. Based on the head orientations, we can interpret the visual search actions of the driver. A detailed technique for identifying the verification of blind spots by the driver is described next.

3. Return to step 1 of the algorithm if we lose one of the facial features.

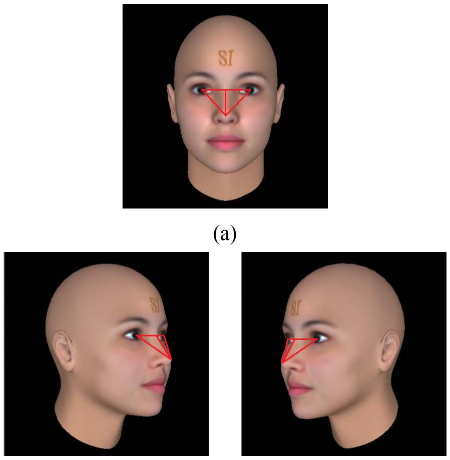


Figure 11. The visual research actions related to the verification of the blind spots. a- Initial position, b- Position due to the verification of the left blind spot, c- Position due to the verification of the right blind spot.

The verification of the blind spot is accompanied by a rotation of the driver's head in the direction of the considered path (cf. Figure 11.(b) and (c)). The angle of rotation established by the driver's head is inversely proportional to the distance separating the two eyes. When the driver verifies the blind spot, the angle of rotation of his head reaches its maximum value. That corresponds to a minimum distance separating the two eyes. Additionally, the coordinates of two eyes allow us to know in which direction the driver is currently watching. Based on these observations, we can accurately identify the visual search action related to the verification of the blind spot. Note that two additional events allow us to identify the verification actions of the blind spots. The first event is the loss of the left eye and/or the nose tip when verifying the left blind spot. The second event is the loss of the right eye and/or the nose tip when verifying the right blind spot. The loss of the nose tip is due to its confusion with the background when verifying the blind spot. While the loss of the eye is due to its partial or total occlusion in the video frame. In Figure 12, we show test results obtained using a cascade of boosted classifier and the pyramidal Lucas-Kanade method for facial features detection and tracking. The loss of nose tip and left eye due to the verification of the left blind spot is shown in Figure 12.(d). The loss of the right eye due to the verification of the right blind spot is shown in Figure 12.(h).

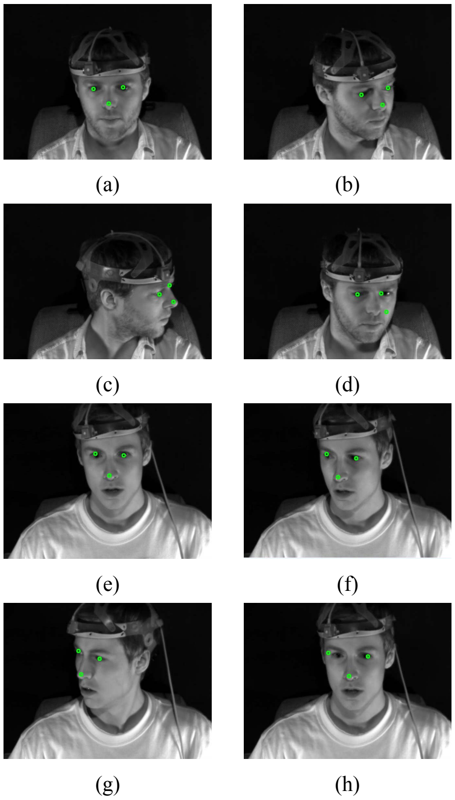


Figure 12. The verification of the blind spots is accompanied by the loss of facial features (nose tip and/or eyes). a,e- First detection of facial features. b,c,f,g- Different steps of the tracking process. d- Loss of nose tip and left eye. h- Loss of the right eye.

In Figure 13, we show test results obtained using the proposed system. Following the loss of facial features, the system triggers the detection process and generates a new event for reporting that the driver is probably verifying the blind spot. Additionally to the study of the cephalo-ocular behavior of driver while verifying the blind spot, the system is able to support other events, such as the visual verifications at rear-view and wing mirrors. Events resulting from the identification and analysis (using our system) of the cephalo-ocular behaviour of drivers will be used by a Kinesiology research group in order to retrain older drivers in a safe-driving context. The introduced system serves as a basis framework for a new system involving three cameras (cf. Figure 14).

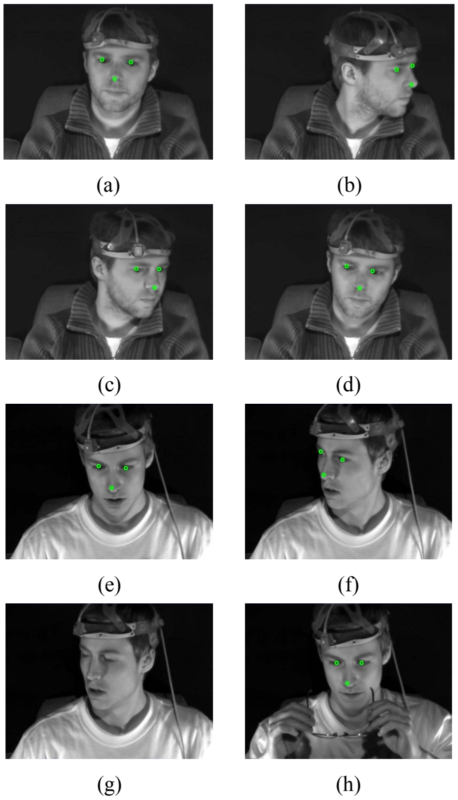


Figure 13. Test results of the introduced framework. a,e- First detection of facial features. b,c,d,f- Different steps of the tracking process. g- The loss of facial features triggers the detection process. h- Tracking of facial features following the second detection.

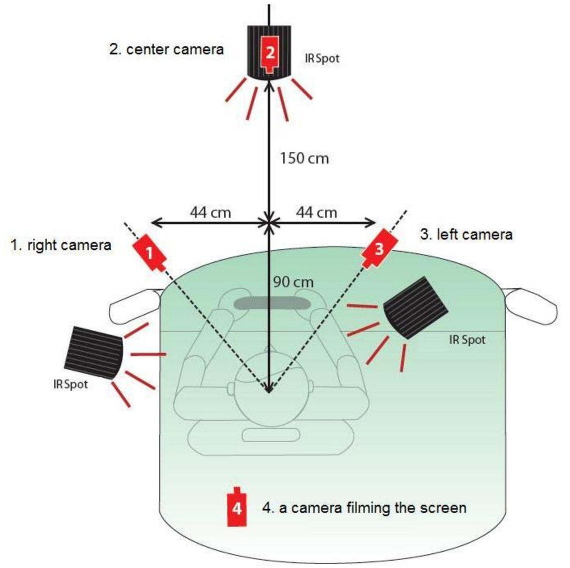
5. Conclusion

In this paper we introduced a new computer vision system dedicated to the analysis and interpretation of the cephalo-ocular behavior of a driver. The proposed system include three main steps. The first step consists in the detection of the most important facial features, namely nose tip and eyes. The detection process is achieved using a cascade of boosted classifiers based on the extended set of haar-like features. The second step deals with the tracking of those facial features. For that we call upon the pyramidal Lucas Kanade method for optical flow estimation. The last step is devoted to the identification of the visual research actions related to the verification of the blind spots using events resulting from the second step. The analysis and interpretation of other visual research actions of driver is also possible to achieve. All of the experiments confirm both the accuracy of the proposed system and its usefulness.

Appendix:

In Figure 14 we show the configuration of our driving simulator. A driver is seated in a mock-up car and uses normal controls, such as steering wheel and clutch. The driving scenario is projected on the screen in front of the driver. The driver's actions are recorded by the simulator system in order to calculates the position of the virtual vehicle car. Three cameras (1, 2 and 3) are used to film the driver while the fourth one is used to film the simulator's screen. Note that the scene is illuminated with three infrared spots.

In our future work we will develop a new system involving the three cameras filming the driver. The main challenge is the tracking of the driver's facial features across the three cameras (left, center and right). That allows the estimation of the pose (position and orientation) of the driver's head.



(a)

Figure 14. Configuration of the driving simulator.

References

[1] M. Bertalmio, G. Sapiro, and G. Randall. Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):733–737, 2000.

[2] G. Bouchard and B. Triggs. A hierarchical part-based model for visual object categorization. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[3] D. Comaniciu, V. Ramesh, and P. Andmeer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:564–575, 2003.

[4] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] J. Kang, I. Cohen, and G. Medioni. Object reacquisition using geometric invariant appearance model. *In International Conference on Pattern Recongnition*, pages 759–762, 2004.

[6] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. *In IEEE International Conference on Image Processing*, pages 900–903, 2002.

[7] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *In International Joint Conference on Artificial Intelligence*, 1981.

[8] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *In Proceedings of the 8th European Conference on Computer Vision*, May 2004.

[9] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, 2001.

[10] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, June 2000.

[11] H. A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. *In Advances in Neural Information Processing Systems*, 8, November 1995.

[12] V. Salari and I. K. Sethi. Feature point correspondence in the presence of occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):87–91, 1990.

[13] K. Sato and J. Aggarwal. Temporal spatio-velocity transform and its application to tracking and interaction. *Computer Vision and Image Understanding*, 96(2):100–128, 2004.

[14] H. Schneiderman and T. Kanade. A statistical model for 3d object detection applied to faces and cars. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[15] H. Schweitzer, J. W. Bell, and F. Wu. Very fast template matching. *In European Conference on Computer Vision*, pages 358–372, 2002.

[16] K. Shafique and M. Shah. A non-iterative greedy algorithm for multi-frame point correspondence. *In IEEE International Conference on Computer Vision*, pages 110–115, 2003.

[17] J. Sklansky. Finding the convex hull of a simple polygon. *Pattern Recognition Letters*, (1):79–83, 1982.

[18] C. Veeman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):54Ű72, January 2001.

[19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[20] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[21] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *In IEEE Conference on Computer Vision and Pattern Recognition*, 2003.