Mercury: a vision-based framework for Driver Monitoring

Guido Borghi[2], Stefano Pini[1], Roberto Vezzani[1], Rita Cucchiara[1]

[1] Dipartimento di Ingegneria Enzo Ferrari
[2] Centro di Ricerca Interdipartimentale Softech-ICT
Università degli Studi di Modena e Reggio Emilia
41125, Modena, Italia
{guido.borghi, s.pini, roberto.vezzani, rita.cucchiara}@unimore.it

Abstract. In this paper, we propose a complete framework, namely *Mercury*, that combines Computer Vision and Deep Learning algorithms to continuously monitor the driver during the driving activity. The proposed solution complies to the requirements imposed by the challenging automotive context: the light invariance, in order to have a system able to work regardless of the time of day and the weather conditions. Therefore, infrared-based images, *i.e.* depth maps (in which each pixel corresponds to the distance between the sensor and that point in the scene), have been exploited in conjunction with traditional intensity images.
Second, the non-invasivity of the system is required, since driver's movements must not be impeded during the driving activity: in this context, the use of cameras and vision-based algorithms is one of the best solutions. Finally, real-time performance is needed since a monitoring system must immediately react as soon as a situation of potential danger is detected.

Keywords: Driver Monitoring · Human-Car Interaction · Computer Vision · Deep Learning · Convolutional Neural Networks · Depth Maps

1 Introduction

The loss of vehicle control is a common problem, due to driving distractions, also linked to driver's stress, fatigue and poor psycho-physical conditions. Indeed, humans are easily distractible, struggling to keep a constant concentration level and showing signs of drowsiness just after a few hours of driving [1].

Furthermore, the future arrival of (semi-)autonomous driving cars and the necessary transition period, characterized by the coexistence of traditional and autonomous vehicles, is going to increase the already-high interest about driver attention monitoring systems, since for legal, moral and ethical implications driver must be ready to take the control of the (semi-)autonomous car [2].

Therefore, in this paper, a driver monitoring system, here referred as *Mercury*, based on Computer Vision and Deep Learning algorithms, is proposed.
From a technical point of view, *Mercury* framework is composed of different stages.

Given an input depth map, in the first step the head center is automatically detected, according to which a bounding box containing the driver's head and a minor part of the background is extracted.
Then, this crop is fed into a system that estimates the 3D pose of the head, in terms of *yaw*, *pitch* and *roll* angles. On the same head crop, a facial landmark localization algorithm is applied, in order to identify the salient areas that belong to the face and are visible in both the depth and the intensity frame.

We believe that head pose information could be useful in terms of driver monitoring, to understand where the driver is looking (*e.g.* inside or outside the car cabin) and for the infotainment systems, in order to make new systems for the *Human-Car Interaction* more *intuitive* and *user-friendly* and to improve the velocity of operations conducted by the driver inside the cockpit [3].

Finally, driver's eyes, detected through the facial landmark system mentioned above, are analyzed to estimate the level of driver drowsiness. Specifically, PERCLOS, a drowsiness detection measure, is exploited.

2 Mercury

The whole architecture of the system is represented in Figure 1. The system has been set up in a modular way, since it is able to work only with RGB data, depth data or both. The functions available for each type of data are different and are described in the following paragraphs.

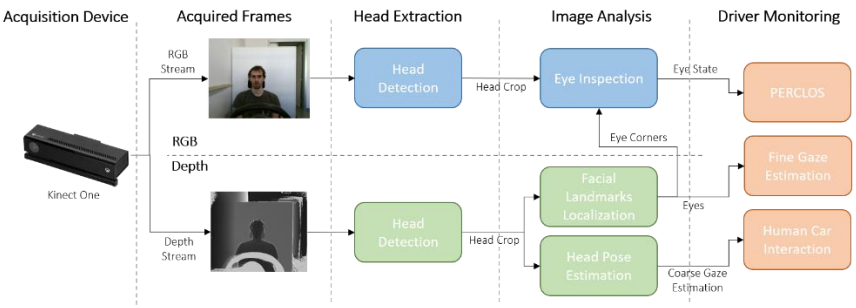

Fig. 1. Overall architecture of the proposed framework. Blue blocks represent algorithms applied on RGB or gray-level frames, while in the green ones are reported the solutions for the depth maps. Finally, orange boxes represent the computed measures for the Driver Monitoring task.

2.1 Acquisition Device

The *Microsoft Kinect One* sensor has been exploited as acquisition device. Although more recent depth sensors are available on the market, with more suitable factor forms for the automotive field, this device has been preferred due to the availability and relative simplicity of the proprietary *SDK* that is compatible with the *Python* programming

language[1], as well as the simultaneous presence of an excellent color camera and a depth sensor with a good spatial resolution and a low noise level. Moreover, the minimum range for data acquisition allows its use on the automotive context, since in our case the device is placed in front of the driver, near the car dashboard.

In the following, technical specifications of the device are reported:

- *Full HD RGB camera*: the spatial resolution is 1920×1080. The device is able to acquire up to 30 frame per second;
- *Depth sensor*: the spatial resolution is 512×424. This sensor provides depth information as a two-dimensional array of pixels, namely *depth map*, similar to a gray-level image. Since each pixel represents the distance in millimeters from the camera, depth images are coded in 16 bits;
- *Infrared emitter*: this device is based on the *Time-of-Flight* (ToF) technology and its range starts from 0.5 up to 7 meters;
- *Microphone array*: in this device is present an array of microphones. Audio data are sampled at 16 kHz and coded with 16 bits. It is used to pick up the voice commands, for the calibration of the peripheral and the suppression of white noise. Microphones are not used in this project.

2.2 Head Extraction

The first step of the framework is the automatic head extraction from the acquired frames. It is assumed that within the acquired scene there is no more than one person, *i.e.* the driver, sitting in front of the acquisition device, about 1 meter away.

For the part of head detection on RGB frames, the well-known object detection framework of *Viola & Jones* [4] has been exploited: despite the presence of more recent deep learning-based algorithms, is still a valid choice for its effectiveness and good speed performance. Moreover, in this regard, we note that many car companies still use derivative versions of the algorithm in question.

Head detection on depth maps is conducted through the *Fully Convolutional Network* proposed in [5]: the limited depth of the network preserves and balances detection accuracy and speed performance.

The output of RGB and depth head detection modules consists of the head crop, which is a bounding box containing the driver's face with minimal background portions.

2.3 Head Pose Estimation

The obtained head crop is used to produce three different inputs for the Head Pose Estimation module. The first input is represented by the raw depth map, while the second is a Motion Image, obtained running the *Optical Flow* algorithm (*Farneback* implementation) on a sequence of depth images. The third input is generated by a network called *Face-from-Depth* [6], that is able to reconstructs gray-level face images starting from the related depth images. All these three inputs are then processed by a regressive *Convolutional Neural Network* (CNN) [7] that finally outputs the value of the yaw, pitch and roll 3D angles expressed as continuous values.

[1] https://github.com/Kinect/PyKinect2

2.4 Facial Landmark Estimation

The goal of this module is a reliable estimation of the facial landmark coordinates, *i.e.* salient regions of the face, like eyes, eyebrows, mouth, nose and jawline. Due to the limited spatial resolution of available depth images, we focus on a selection of five facial landmarks: eye pupils, mouth corners and the nose tip. Accordingly, the system outputs 10 image coordinates, *i.e.*, the x and y values for each facial landmark.

The core of the method is a CNN [8] that works in regression and receives a stream of depth images as input. The ground truth annotation of the landmark positions is required during the network training step and is used as a comparison during the test.

The developed system has real-time performance and it is more reliable than state-of- art competitors in presence of poor illumination and light changes, thanks to the use of depth images as input. The extraction of the facial landmarks allows obtaining the bounding box of the driver's right and left eye: these images are then analyzed by a CNN to determine if the eye is open or closed.

2.5 Driver Monitoring

The analysis of the attention and the level of fatigue of the driver is done through the indicator called PERCLOS (*PERcentage of eyelid CLOSure*), introduced in 1994 in [9]. This measure aims to express the percentage of time in a minute in which the eye remains closed from 80% to 100%.

In general, the computation of the PERCLOS measurement yields a numerical value that can be analyzed to determine the driver's fatigue level. Blinking beats, identifiable as almost instantaneous closures of the eye, are excluded in its computation, since only the prolonged and slow closures, usually called *droops*, are maintained.
The level of driver attention is then classified through three thresholds:

- *Alert* (PERCLOS < 0:3): good level of attention, optimal driver conditions;
- *Drowsy* (0.3 < PERCLOS < 0.7): first signs of carelessness and fatigue, the driver is not in optimal conditions and that introduces some risks;
- *Unfocused* (PERCLOS > 0.7): complete lack of attention of the driver who is in poor physical condition or is sleeping;

3 Implementation

In the following paragraphs, details about the hardware and the software implementation of *Mercury* are reported and discussed.

3.1 Hardware Implementation

The *Mercury* framework has been implemented and tested on a computer equipped with an *Intel Core i7-7700K* processor, 32 GB of RAM and a *Nvidia 1080Ti* GPU.

The various components belonging to the deep learning tools have been implemented with the *Keras*[2] framework with *Tensorflow*[3] as backend.
Finally, a graphic interface has been created through the *Qt Libraries*[4] and it is detailed in the following paragraph.
Being aware that the aforementioned hardware equipment is not suitable for the automotive context, due to reasons related to the costs and power consumption, we have started to implement these systems on embedded boards. The *Nvidia TX2* board has been chosen, since it is equipped with an embedded GPU suitable for deep learning-based algorithms.

3.2 Graphical User Interface

The *Graphical User Interface* (GUI), shown in Figure 2, has been created using the *Qt libraries 4* and *Python* programming language. It has been divided into four vertical sections: 1. Input Visualization, 2. Head Detection, 3. Head Pose Estimation and Facial Landmark Detection and 4. Driver Attention Analysis.

From the left, the first part contains the RGB and depth frames captured by the *Kinect One* device, for visualization and debugging purposes. The side column is dedicated to display the output of the head detection modules. In particular, the localized bounding boxes are shown in green on the frames.
The central part is dedicated to the driver's face analysis, and it is divided into several horizontal sections. Starting from the top, the head pose is displayed either through a 3D cube or with bars, centered on the zero angle. The yaw angle is shown in blue, the roll angle in green and the pitch in red.
Further down, the facial landmarks found on the RGB frame are shown through red dots. Thanks to these facial landmarks, the bounding boxes containing the right eye, the left eye, all displayed at the bottom, are extracted to compute the PERCLOS measure.

At the extreme right, the information regarding the analysis of driver attention are reported. The PERCLOS measure is indicated by its percentage value and, according to the established thresholds, the term alert, drowsy or unfocused is highlighted. The head pose is instead indicated with the representation of the possible positions of the gaze on a given object.
An image of a generic car interior has been then used to graphically convey the concept of *Human-Car Interaction*, highlighting with the red color the object inside the passenger compartment. Some common areas have been chosen as target for the interaction: the right, central and left rear mirrors, the area of the steering wheel, the gearbox and the part of the infotainment.

Finally, in the bottom part of the GUI, the buttons that allow the user to select the stream of data on which the framework must work, *i.e.* RGB, depth or both streams, are placed. Moreover, the measure of the frame per second for each part of the system is shown.

---

[2] https://keras.io/
[3] https://www.tensorflow.org/
[4] https://www.qt.io/

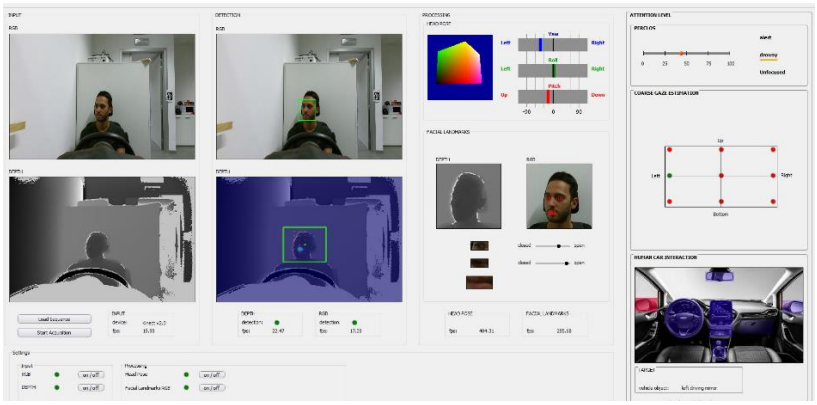

Fig. 2. The Graphical User Interface developed for the *Mercury* framework.

4 Conclusions

In this paper, *Mercury*, a framework for automatic Driver Monitoring, has been introduced. Input data is represented by both RGB images and depth maps, in order to improve the light invariance of the system. Future work will include the implementation of the framework in real embedded boards, suitable for the automotive context.

References

1. Young, K., Regan, M., Hammer, M.: Driver distraction: A review of the literature. Distracted driving, pp. 379-405 (2007)
2. Venturelli, M., Borghi, G., Vezzani, R., Cucchiara, R.: Deep head pose estimation from depth data for in-car automotive applications. In International Workshop on Understanding Human Activities through 3D Sensors, pp. 74-85 (2016)
3. Nawaz, T., Mian, M. S., Habib, H. A.: Infotainment devices control by eye gaze and gesture recognition fusion. In IEEE Transactions on Consumer Electronics, 54(2), pp. 277-282 (2008)
4. Viola P., Jones M. J.: Robust real-time face detection. In International Journal of Computer Vision, vol. 57, no. 2, pp. 137-154 (2004)
5. Ballotta, D., Borghi, G., Vezzani, R., Cucchiara, R.: Fully Convolutional Network for Head Detection with Depth Images. In 24th International Conference on Pattern Recognition (ICPR), pp. 752-757 (2018)
6. Borghi, G., Fabbri, M., Vezzani, R., Cucchiara, R.: Face-from-Depth for Head Pose Estimation on Depth Images. In IEEE transactions on pattern analysis and machine intelligence (2018)
7. Borghi, G., Venturelli, M., Vezzani, R., Cucchiara, R.: Poseidon: Face-from-depth for driver pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4661-4670 (2017)
8. Frigieri, E., Borghi, G., Vezzani, R., Cucchiara, R.: Fast and accurate facial landmark localization in depth images for in-car applications. In International Conference on Image Analysis and Processing, pp. 539-549 (2017)
9. Wierwille, W. W., Wreggit, S. S., Kirn, C. L., Ellsworth, L. A., Fairbanks, R. J.: Research on vehicle-based driver status/performance monitoring; development, validation, and refinement of algorithms for detection of driver drowsiness (1994)