

PublisherInfo	
PublisherName	: Springer International Publishing
PublisherLocation	: Cham
PublisherImprintName	: Springer

## Driver Drowsiness Warning System Using Visual Information for Both Diurnal and Nocturnal Illumination Conditions

ArticleInfo									
ArticleID	: 2794								
ArticleDOI	: 10.1155/2010/438205								
ArticleCitationID	: 438205								
ArticleSequenceNumber	: 136								
ArticleCategory	: Research Article								
ArticleCollection	: Vehicular Ad Hoc Networks								
ArticleFirstPage	: 1								
ArticleLastPage	: 1								
ArticleHistory	: <table border="0"> <tr> <td>RegistrationDate</td> <td>: 2009-11-23</td> </tr> <tr> <td>Received</td> <td>: 2009-11-23</td> </tr> <tr> <td>Accepted</td> <td>: 2010-6-21</td> </tr> <tr> <td>OnlineDate</td> <td>: 2010-7-11</td> </tr> </table>	RegistrationDate	: 2009-11-23	Received	: 2009-11-23	Accepted	: 2010-6-21	OnlineDate	: 2010-7-11
RegistrationDate	: 2009-11-23								
Received	: 2009-11-23								
Accepted	: 2010-6-21								
OnlineDate	: 2010-7-11								
ArticleCopyright	: Marco Javier Flores et al.2010 This article is published under license to BioMed Central Ltd. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.								
ArticleGrants	:								
ArticleContext	: 136342010201011								

MarcoJavier Flores,<sup>Aff1</sup>  
 Email: mjflores@ing.uc3m.es  
 JoséMaria Armingol,<sup>Aff1</sup>  
 Email: armingol@ing.uc3m.es  
 Arturo de la Escalera,<sup>Aff1</sup>  
 Corresponding Affiliation: Aff1  
 Email: escalera@ing.uc3m.es

Aff1 University Carlos III of Madrid, C/. Butarque 15, 28991 Leganés, Madrid, Spain

### Abstract

Every year, traffic accidents due to human errors cause increasing amounts of deaths and injuries globally. To help reduce the amount of fatalities, in the paper presented here, a new module for Advanced Driver Assistance System (ADAS) which deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence is presented. The aim of this system is to locate, track, and analyze both the drivers face and eyes to compute a drowsiness index, where this real-time system works under varying light conditions (diurnal and nocturnal driving). Examples of different images of drivers taken in a real vehicle are shown to validate the algorithms used.

## 1. Introduction

ADAS is part of the active safety systems that interact to a larger extent with drivers to help them avoid traffic accidents. The goal of such systems is to contribute to the reduction of traffic accidents by means of new technologies; that is, incorporating new systems for increasing vehicle security, and at the same time, decreasing danger situations that may arise during driving, due to human errors. In this scenario, vehicle security research is focused on driver analysis. In this particular research, a more in-depth analysis of drowsiness and distraction is presented [1].

Drowsiness appears in situations of stress and fatigue in an unexpected and inopportune way and may be produced by sleep disorders, certain types of medications, and even, boredom, for example, driving for long periods of time. The sleeping sensation reduces the level of vigilance producing danger situations and increases the probability of an accident occurring.

It has been estimated that drowsiness causes between 10% and 20% of traffic accidents, causing both fatalities dead [2] and injuries [3], whereas within the trucking industry 57% of fatal truck accidents are caused by this problem [4, 5]. Fletcher et al. in [6] have stated that 30% of all traffic accidents have been caused by drowsiness, and Brandt et al. [1] have presented statistics showing that 20% of all accidents are caused by fatigue and lack of attention. In the USA, drowsiness is responsible for 100000 traffic accidents yearly producing costs of close to 12.000 million dollars [7]. In Germany, one out of four traffic accidents originate from drowsiness, while in England 20% of all traffic accidents are produced by drowsiness [8], and in Australia 1500 million dollars has been spent on fatalities resulting from this problem [9].

In this context, it is important to use new technologies to design and build systems that are capable of monitoring drivers and to measure their level of attention during the complete driving process. Fortunately, people in a state of drowsiness produce several visual cues that can be detected in the human face, such as

- (i) [yawn frequency](#),
- (ii) [eye-blinking frequency](#),
- (iii) [eye-gaze movement](#),
- (iv) [head movement](#),
- (v) [facial expressions](#).

By taking advantage of these visual characteristics, computer vision is the most feasible and appropriate technology available to deal with this problem. This paper presents the drowsiness detection system of the IVVI (Intelligent Vehicle based Visual on Information) vehicle [10]. The goal of this system is to automatically estimate the driver's drowsiness and to prevent drivers falling asleep at the wheel.

This paper is laid out as follows. Section 2 presents an extensive review on the state of the art considering different lighting conditions. A general framework of the proposed method is presented Section 3. There are two systems, one for diurnal and another nocturnal driving. Both have a first step for face and eye detection, followed for a second step for face and eye tracking. The output of both systems is a drowsiness index based on a support vector machine. A deeper explanation of both systems is presented in Sections 4 and 5 where the similarities and differences of both approaches are highlighted, and the results are shown. Finally, in Section 6, the conclusions are presented.

## 2. Related Work

To increase traffic safety and to reduce the number of traffic accidents, numerous universities, research centers, automotive companies (Toyota, Daimler Chrysler, Mitsubishi, etc.), and governments (Europe Union, etc.) are contributing to the development of ADAS for driver analysis [5], using different technologies. In this sense, the use of visual information to obtain the state of the driver drowsiness and to understand his/her behavior is an active research field.

This problem requires the recognition of human behavior when in a state of sleepiness by means of an eye and facial (head) analysis. This is a difficult task, even for humans, because there are many factors involved, for instance, changing illumination conditions and a variety of possible facial postures. Considering the illumination, the state of the art has been divided in two parts: the first provides details on systems that work with natural daylight whereas the second deals with systems which operate with the help of illumination systems based on near infrared (NIR) illumination.

### 2.1. Systems for Daylight Illumination

To analyze driver drowsiness several systems have been built in recent years. They usually require the problem to be simplified to work partially or for specific environments: for example, D'Orazio et al. [11] have proposed an eye detection algorithm that searches for the eyes within the complete image and have assumed that the iris is always darker than the sclera. Using the Hough transform for circles and geometrical constraints the eye candidates are located; next, they are passed to a neural network that classifies between eyes and noneyes. This system is capable of classifying eyes as being open or closed. The main limitations to this algorithm are as follows. It is applicable only when the eyes are visible in the image, and it is not robust for changes in illumination. Hornig et al. [12] have presented a system that uses a skin color model over an HSI space for face detection, edge information for eye localization, and dynamical template matching for eye tracking. By using color information from the eyeball, the state of the eye is defined; thus the driver's state can be computed, that is, asleep or alert; if the eyes are closed for five consecutive frames, the driver is assumed to be dozing. Brandt et al. [1] have shown a system that monitors driver fatigue and lack of attention. For this task, the Viola Jones (VJ) method has been used [13] to detect the driver's face. By using the optical flow algorithm on eyes and the head this system is able to compute the driver's state. Tian and Qin in [2] have built a system which verifies the state of the driver's eye. Their system uses Cb and Cr components of the YCbCr color space: with a vertical projection function this system locates the face region and with a horizontal projection function it locates the eye region. Once the eyes are located the system computes the eye state using a complexity function. Dong and Wu [3] have presented a system for driver fatigue detection: this is based on a skin color model on a bivariate Normal distribution and Cb and Cr components of the YCbCr color space. After locating the eyes, it computes the fatigue index using the distance of the eyelid to classify whether the eyes are open or closed; if the eyes are closed for five consecutive frames, the driver is considering to be dozing, as in Hornig's work. Branzan et al. [14] also have presented a system for drowsiness monitoring using template matching to analyze the state of the eye.

### 2.2. Systems Using Infrared Illumination

As a result of nocturnal lighting conditions, Ji et al. in [4, 15] have presented a drowsiness detection system based on NIR illumination and stereo vision. This system locates the position of the eye using image differences based on the bright pupil effect. Later, this system computes the blind eyelid frequency and eye gaze to build two drowsiness indices: PERCLOS (percentage of eye closure over time) [7] and AECS (average eye closure speed). Bergasa et al. [5] have also developed a nonintrusive system using infrared light illumination this system computes the driver's vigilance level using a finite state automata (FSM) [16] with six different eye states that compute several indices, among them, PERCLOS; this system is also capable of detecting inattention considering a facial posture analysis. Other research work based on this type of illumination has been presented by Grace [17], where the authors measure slow eyelid closure. Systems using NIR illumination work well under stable lighting conditions [5, 18]; however, these systems present drawbacks for applications in real vehicles, where the light continually changes. In this scenario, if the spectral pupils disappear, then the eye detection process becomes more complex.

## 3. System Design for Drowsiness Detection

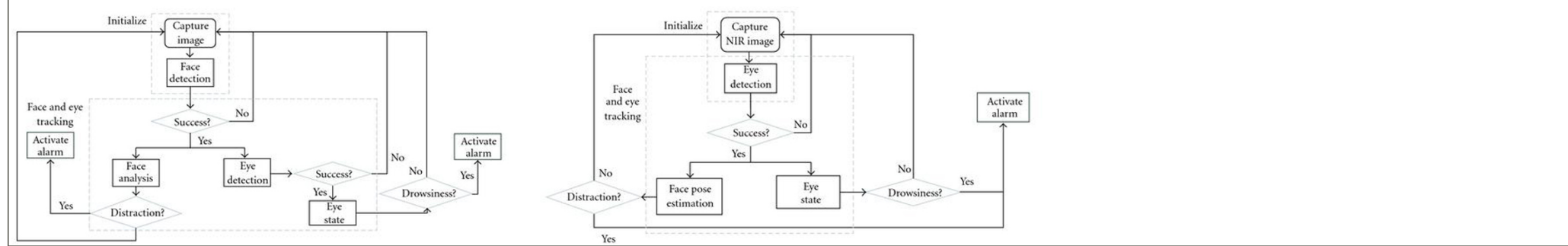
This paper presents a system which detects driver drowsiness which works for both day and night time conditions and follows the classification presented in the state of the art.

This composition has allowed two systems to be obtained, one for day and a second for night time conditions. The first works with natural daylight illumination and the second with artificial infrared illumination. It is interesting to note that both systems operate using grayscale images taken within a real vehicle.

The general scheme of both systems is shown in Figure 1, where six modules are presented as follows:

- (i) face detection;
- (ii) eye detection;
- (iii) face tracking;
- (iv) eye tracking;
- (v) drowsiness detection;
- (vi) distraction detection.

Figure 1 Algorithm scheme: (a) day system, (b) night system.



Each one of these parts will be explained in the following sections.

## 4. Day System Design

In this section, the daytime system based on the algorithm schematic shown in Figure 1(a) will be described, where the visual information is acquired using a digital camera.

### 4.1. Face Detection

To locate the face, this system uses the VJ object detector which is a machine learning approach for visual object detection. This makes use of three important features to make an efficient object detector based on the integral image, the AdaBoost technique and the cascade classifier [13]. Each one of these elements is important to efficiently process the images and in near real-time with correct detections as high as 90%. A further important aspect of this method is its robustness for changing light conditions. However, in spite of the above-mentioned features, its principal disadvantage is that it cannot extrapolate and does not work appropriately when the face is not in front of the camera axis. This particular case occurs when the driver moves his/her head. This shortcoming will be analyzed later on in this paper.

Continuing with the algorithm description, when the driver's face is detected, it is enclosed within a rectangle RI (region of interest) which is defined by its left-top corner coordinates  $P_0 = (x_0, y_0)$  and bottom-right corner coordinates  $P_1 = (x_1, y_1)$ , as can be observed in Figures 2(a), 2(b), and 2(c). The size of the rectangle has been determined from experimental analysis developed on the face database that has been created for this task.

Figure 2 Viola Jones method.



## 4.2. Eye Detection

Locating the position of the eye is a difficult task as different features define the same eye depending, for example, on the area of the image where it appears and on the color of the iris, but the main problem that occurs when driving is the changes in the ambient lighting conditions.

Once the face has been located within the rectangle  $RI$  described in the previous section, using the face anthropometric properties [19] which are derived from a face database analysis, two rectangles containing the eyes are obtained. This system uses  $RI_L$  for the left eye rectangle and  $RI_R$  for the right eye rectangle; this is shown in the following four equations and also in Figure 3:

$$\begin{aligned} (u_{0L}, v_{0L}) &= \left(x_0 + \frac{w}{6}, y_0 + \frac{h}{4}\right) \\ (u_{1L}, v_{1L}) &= \left(x_0 + \frac{w}{2}, y_0 + \frac{h}{2}\right) \\ (u_{0R}, v_{0R}) &= \left(x_0 + \frac{w}{2}, y_0 + \frac{h}{4}\right) \\ (u_{1R}, v_{1R}) &= \left(x_1 - \frac{w}{6}, y_1 - \frac{h}{2}\right) \end{aligned}$$

(1)

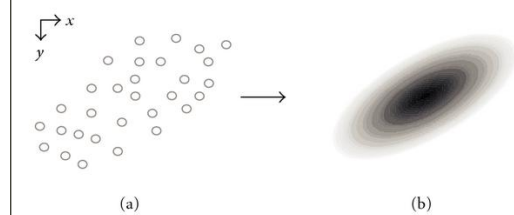
where  $w = x_1 - x_0$  and  $h = y_1 - y_0$ .

Figure 3 Eye rectangles  
and



After the previous step: the exact position of each eye is searched for by incorporating information from the grey-level pixels. The main idea here is to obtain a random sample from the pixels that belong to the eye area, and then, to adjust a parametric model. Figure 4 shows this procedure where a random sample has been extracted in (a), and an elliptical model has been adjusted in (b). In this case, the eye state is independent, that is, it may be open or closed.

Figure 4 (a) Random sample, (b) eye parametric model.



To extract the random sample, the following algorithm has been proposed. Let  $I(x, y) \in [0, 255]$  be the pixel value at the position  $(x, y)$ , then do as follows.

- (i) Generate image  $J$  using the following:

$$J(x, y) = \frac{I(x, y) - m}{\sigma},$$

where  $m$  and  $\sigma$  are the mean and the standard deviation, respectively. These parameters are computed over the previously located eye rectangles.

(2)

(ii) Generate image  $K$  using

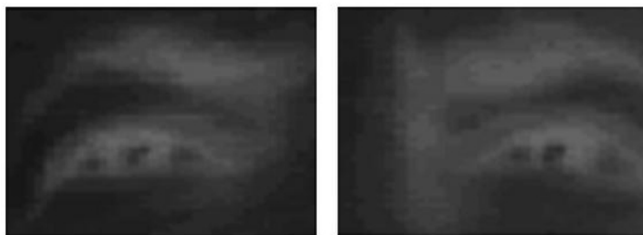
$$K(x, y) = \begin{cases} \lambda(x, y) - 256 * \delta_1 & \text{if } \lambda(x, y) \geq 0, \\ 256 * \delta_2 + \lambda(x, y) & \text{if } \lambda(x, y) < 0, \end{cases}$$

where  $\delta_1 = \max(0, \text{ceil}(\lambda(x, y) / 256) - 1)$ ,  $\delta_2 = \max(1, \text{ceil}(|\lambda(x, y)| / 256))$ , and  $\text{ceil}(x)$  is the function that returns the smallest integer larger than  $x$ .

- (iii) Obtain the binary image,  $B$ , from image  $K$  using Ostu's method [20] which calculates an automatic threshold (Figure 5(b)).
- (iv) Compute the gradient image,  $G$ , using the Sobel horizontal ( $\hat{S}_x$ ) and vertical ( $\hat{S}_y$ ) edge operator followed by an image contrast enhancement [21] (Figure 5(c)).
- (v) Compute the logarithm image [22],  $L$ , where the objective here is to enhance the iris pixels that form the central part of the eye (Figure 5(d)).

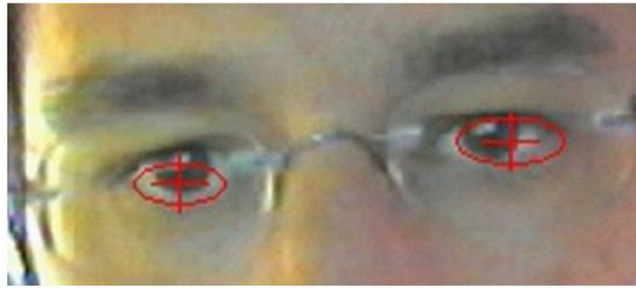
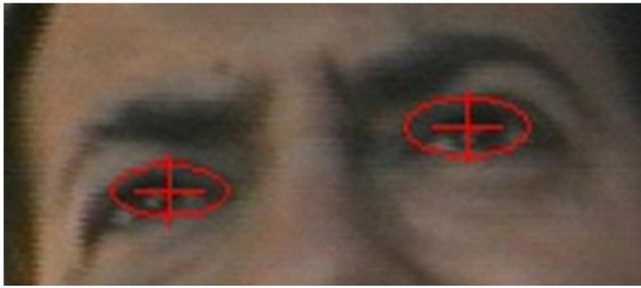
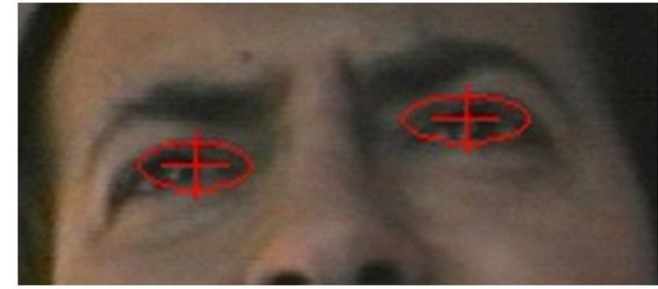
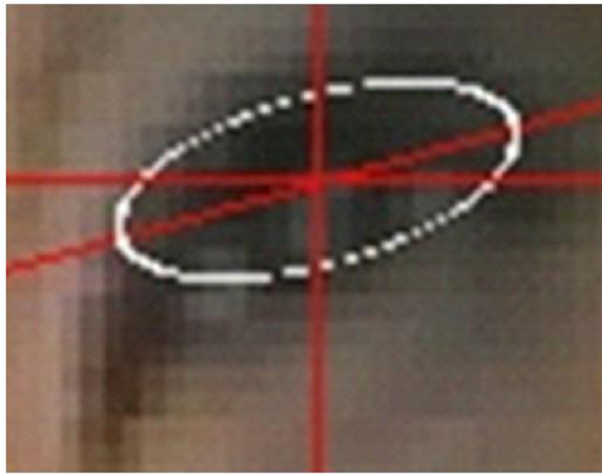
Figure 5 Eye location using [\[MediaObjects/13634\\_2009\\_Article\\_2794\\_IEq23\\_HTML.gif\]](#)

and [\[MediaObjects/13634\\_2009\\_Article\\_2794\\_IEq24\\_HTML.gif\]](#) (a) grayscale image, (b) binary image (B), (c) gradient image (G), and (d) logarithm image (L).



With the pixels that have been extracted from images B, G, and L, it is possible to obtain the previously mentioned random sample. This sample presents an ellipse shape where an elliptical model has been adjusted over this using the expectation maximization algorithm (EM) [23]. Special attention is paid to the center of the ellipse, because, it allows the exact position of the center of the eye center to be obtained. The ellipse axes determine the width and height of the eyes. The result is shown in Figure 6(b).

Figure 6 Expectation maximization algorithm over the spatial distribution of the eye pixels: (a) eye image, (b) ellipse parameters: center, axes, and inclination angle. (c), (d), (e), and (f) show other examples of this procedure.



The main reason behind using pixel information from a random sample is due to the fact that head movements, illumination changes, and so forth, do not allow complete eye pixel information to be obtained, that is, only partial information of the eye in images B, G, and L is available, where the elliptical shape prevails. This random information makes it feasible to use an algorithm that computes the parameters of a function which approximate the eye ellipse shape. EM computes the mean, variance, and the correlation of the  $X$  and  $Y$  coordinates that belong to the eye. The initial parameters required to run the EM are obtained from a regression model adjusted using the least square method. The number of iterations of the EM algorithm is set to 10, and the sample size is taken to be at least  $1/3$  of the rectangles area  $R_{lr}$ . These parameters will be used in the eye state analysis presented below.

### 4.3. Tracking

There are a number of reasons for using a tracking module. The first is due to problems that were encountered using the VJ during this research. Another is related with the necessity to track the face and eyes continuously from frame to frame. A third reason is to reduce the search space thus satisfying the real-time condition requirement. The tracking process has been developed using the Condensation Algorithm (CA) in conjunction with Neural Networks (NNs) used for face tracking and with template matching for eye tracking.

#### 4.3.1. The Condensation Algorithm

This contribution implements the CA that was proposed by Isard and Blake [24, 25] to track active contours using a stochastic approach. CA combines factored sampling (Monte-Carlo sampling method) with a dynamic model that is governed by the state equation

$$X_t = f(X_{t-1}, \xi_t),$$

(4)

where  $X_t$  is the state at instant  $t$  and  $f(\cdot)$  is a nonlinear equation and depends on the previous state plus a white noise. The goal here is to estimate the state vector  $X_t$  with the help of system observation, which are the realization of the stochastic process  $Z_t$  governed by the measurement equation

$$Z_t = h(X_t, \eta_t),$$

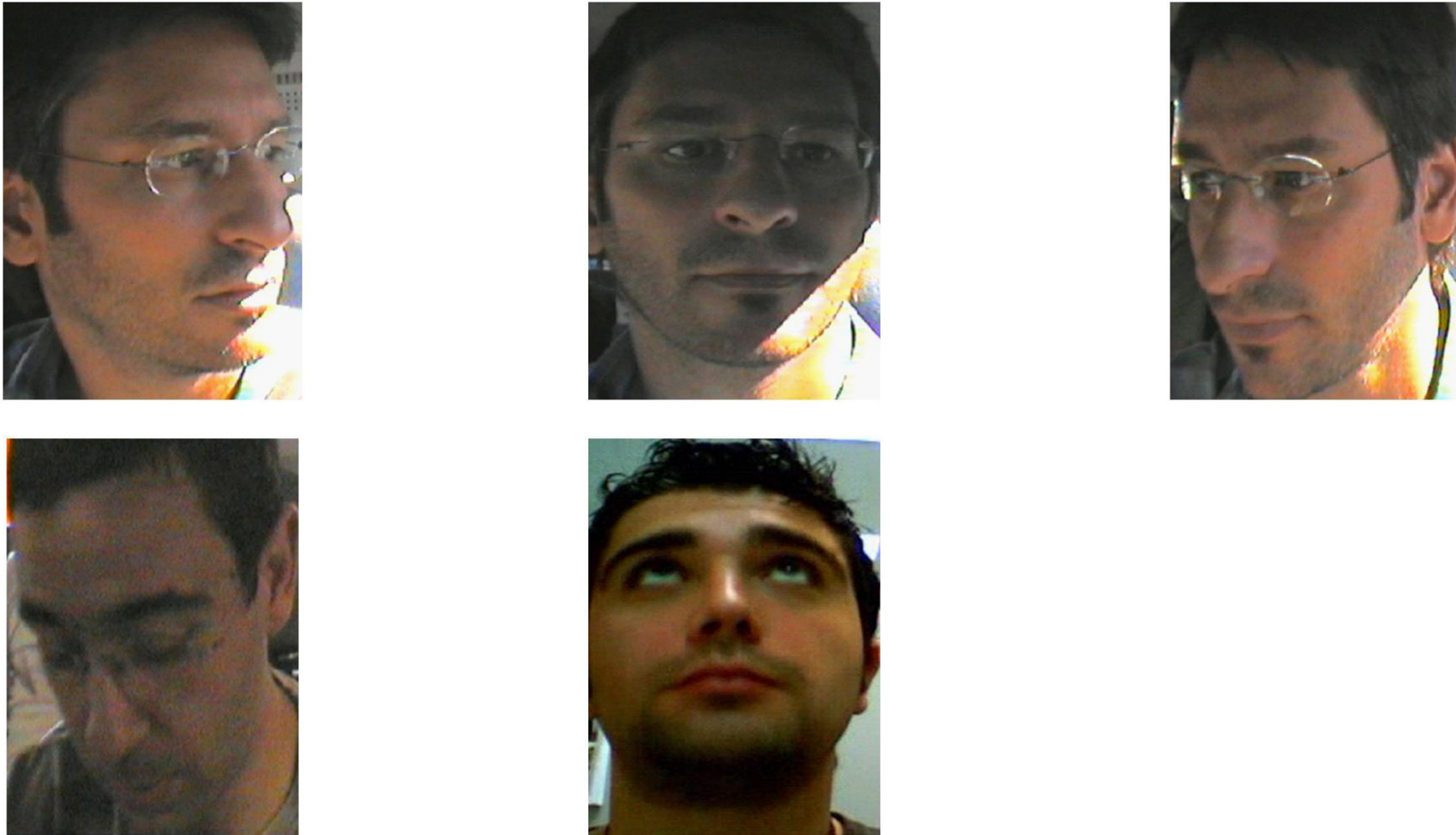
(5)

where  $Z_t$  is the measurement system at time  $t$  and  $h(\cdot)$  is a nonlinear equation that links the present state plus a white noise. The processes  $\xi_t$  and  $\eta_t$  are both white noise terms and are independent of each other. Also, these processes in general are non-Gaussian and multimodal. It must be pointed out that  $X_t$  is an unobservable underlying stochastic process.

### 4.3.2. Neural Networks

Neural networks are used in a wide variety of pattern recognition and classification problems. Figure 7 shows several face examples used to train the backpropagation neural network used in this algorithm.

Figure 7 Examples of a face database which contain faces with different orientations: (a) left profile, (b) front view, (c) right profile, (d) down profile, and (e) up profile.



Before the neural network is trained, a two-part preprocessing step is necessary.

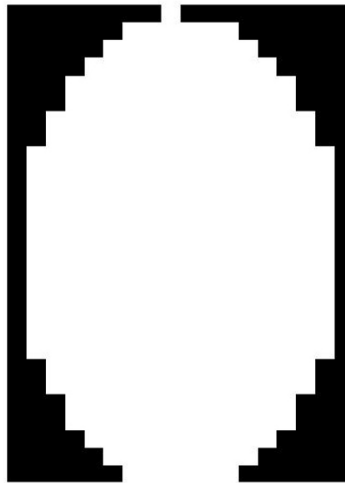
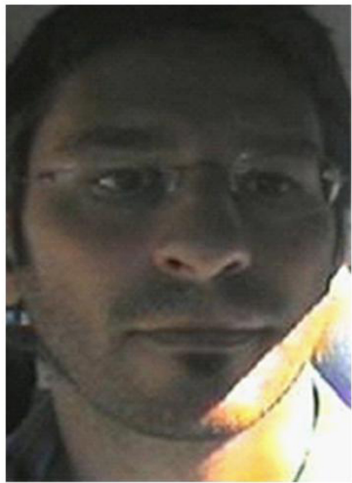
- (i) Contrast modification using gamma correction given by (6) with  $\gamma = 0.8$  where this value has been determined experimentally [26]

$$I(x, y) = I(x, y)^\gamma.$$

- (ii) Remove the contour points by means of a masked AND operation shown in Figure 8(a).

(6)

Figure 8 Mask for face training and its result.

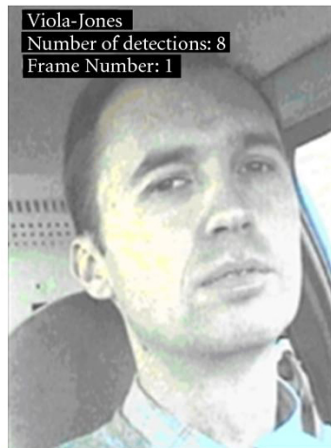


Next, the characteristic vector which consists of the pixel gray-level values from the face image is extracted. The rate of classification following training is greater than 93%.

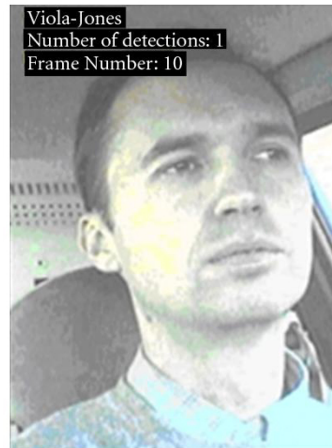
### 4.3.3. Face Tracking

Previously, it has been mentioned that the VJ method presents problems when detecting faces that deviate from the nominal position and for different orientations; thus, to correct this disadvantage a face tracking method has been developed. To demonstrate this shortcoming, Figure 9 shows several different time instants where the VJ method is not capable of finding the driver's face. Figure 10 presents an extended example, where the true position and the VJ position are represented over a frame sequence. The true position has been retrieved manually.

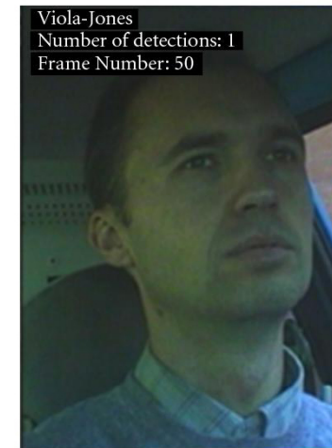
Figure 9 The driver's face has not been found by the Viola Jones method during several time instants.



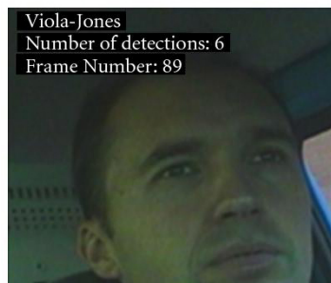
$t = 1$



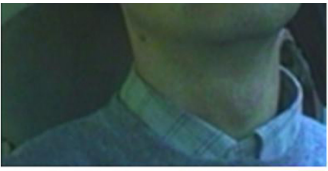
$t = 10$



$t = 50$

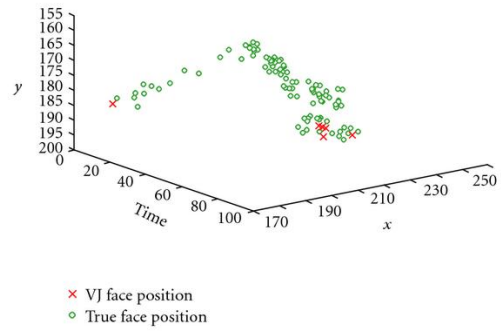






$t = 89$

Figure 10 Example where the VJ method does not find the driver's face in a 100-frame sequence.



The main problem of the VJ method is that it is only able to locate the human face when it is positioned in front of the camera. This drawback leads to an unreliable system for driver analysis throughout the driving process which is highly dynamic, for example, when looking at the rearview or wing mirrors. Much effort has gone into correcting this problem resulting in an efficient tracker which has been implemented using CA combined with a backpropagation neural network.

Through recursive probabilistic filtering of the incoming image stream, the state vector

$$X_t = (x_c, y_c, u_c, v_c, w, h)^T \in R^6 \quad (7)$$

for a driver's face is estimated for each time step  $t$ . It is characterized by its position, velocity, and size. Let  $(x_c, y_c)$  represent its center position,  $(u_c, v_c)$  its velocity in both  $x$  and  $y$  directions and  $(w, h)$  the size in pixels. The measurement vector is given by

$$Z_t = (x_c, y_c, w, h)^T \in R^4 \quad (8)$$

The dynamics of the driver's face has been modeled as a second-order autoregressive process  $AR(2)$ , according to

$$X_t = A_2 X_{t-2} + A_1 X_{t-1} + \xi_t \quad (9)$$

where  $A$  is the transition matrix proposed in [25], and  $\xi_t$  represents the system perturbation at time  $t$ . The most difficult part of the CA is to evaluate the observation density function. In this contribution to compute the weight  $\hat{\pi}_t^{(j)} = p(z_t | x_t = s_t^{(j)})$  for  $j = 1, \dots, N_t$  at time  $t$ , a neural network value in the range of  $[0, 1]$  has been used; this provides an approximation of the face and nonface in conjunction with the distance and with respect to the face to track. This is similar to the work performed by Satake and Shakunaga [27] who have used sparse template matching to compute the weight  $\hat{\pi}_t^{(j)}$  of the sample  $s_t^{(j)}$  for  $j = 1, \dots, N_t$ . In this contribution, the neural network value is used as an approximate value for the weights.

The density function of the initial state is  $p(x_0) = N(x_0, \hat{x}_0)$ , where  $\hat{x}_0$  is computed using the VJ method, and  $\hat{x}_0$  is given in [4]. A particle representation at time  $t$  is shown in Figures 11(a) and 11(b) and Figure 12 depicts the tracking process in which the green circle is the true position and the red cross characterizes a particle or hypothesis, whereas Figure 13 shows the probability over time. This tracker is highly flexible as the neural network includes faces and nonfaces for different head orientations under various illumination conditions. Table 1 presents further results for several sequences of drivers faces. The sequences come from the driver database, which was created to perform these experiments. The true face position has been retrieved manually.

Table 1 Result of face tracking.

Driver	Total frames	Tracking failure	Correct rate
D1	960	60	93.75%
D2	900	22	97.55%
D3	500	45	91.00%
D4	330	15	95.45%
D5	1400	50	96.42%

Figure 11 One time step of the Condensation algorithm (a) predicted region, (b) particles regions.

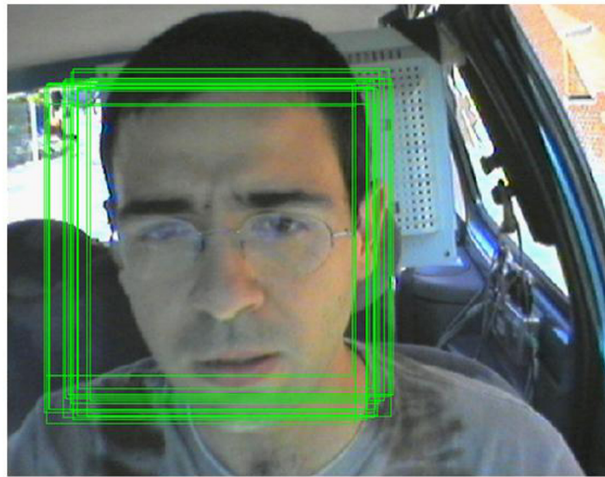


Figure 12 Trajectory of the real and estimated face-center in a 100-frame sequence using the proposed tracker.

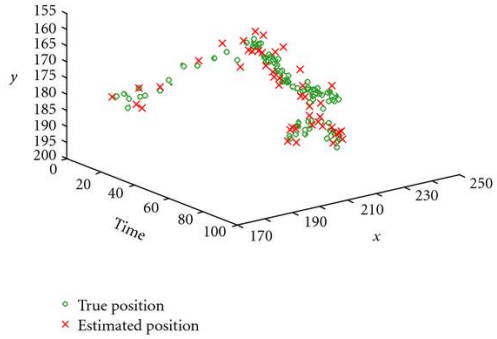
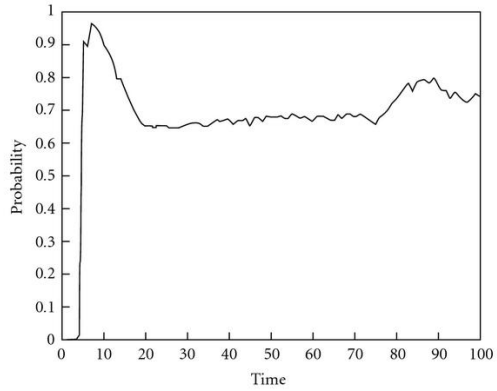


Figure 13 Estimated value of the a posteriori density of the face-center in a 100-frame sequence using the proposed tracker; the face is detected in the fourth frame.



#### 4.3.4. Eye Tracking

For this task, the state of the eye is characterized by its position and velocity over the image. Let  $(x, y)$  represent the eye pixel position at time  $t$  and  $(u, v)$  be its velocity at time  $t$  in the  $x$  and  $y$  directions, respectively. The state vector at time  $t$  can, therefore, be represented by

$$X_t = (x, y, u, v)^T \in \mathbb{R}^4. \tag{10}$$

The transition model is given by (11) which is a first autoregressive model AR(1)

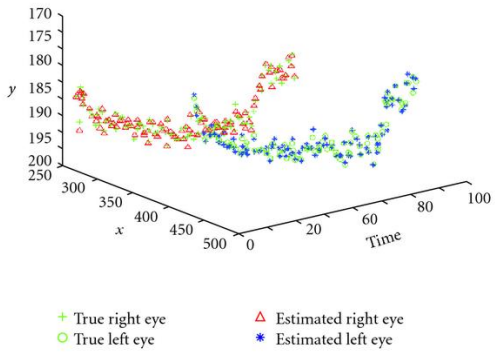
$$X_t = AX_{t-1} + \xi_t. \tag{11}$$

The evaluation of the observation density function is developed by a template matching strategy [27] that was truncated to reduce false detections. CA is initialized when the eyes are detected with the method described in the previous section plus a white noise, that is, similar to the face tracking case. Figure 14 depicts the eye trajectory tracking and Figure 15 shows the compute value of the a posteriori density function of each eye, both on a sequence of 100 images. Table 2 shows the eye tracking results that have been obtained from several sequences of images during the experiments.

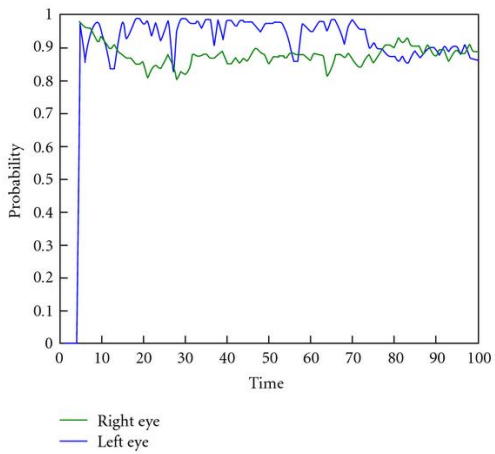
**Table 2** Result of eye tracking.

Driver	Total frames	Tracking failure	Correct rate
D1	960	20	97.91%
D2	900	30	96.60%
D3	500	8	98.40%
D4	330	14	95.75%
D5	1400	90	93.57%

**Figure 14** Trajectory of the real and estimated eyes center in a 100-frame sequence.



**Figure 15** Estimated value of the a posteriori density of the eye center in a 100-frame for right and left eyes; the eyes are detected in the fourth frame.

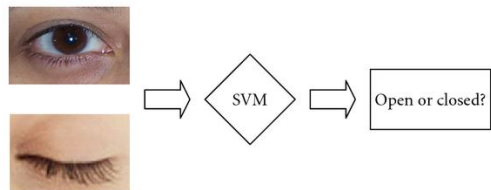


## 4.4. Eye State Detection

To identify drowsiness using eye analysis it is necessary to know its state, that is, open or closed, through the time and to develop an analysis over time, that is, to measure the time that has passed for each state. Classification of the open and closed state is complex due to the changes in the shape of the eye, among other factors, the changing position and the face rotations and variations of twinkling and illumination.

All of these problems make it difficult to reliably analyze the eye. To overcome these shortcomings a supervised classification method has been used, more specifically, a Support Vector Machine (SVM). Figure 16 presents the proposed scheme for eye state verification.

Figure 16 SVM scheme for eye state verification.



#### 4.4.1. Support Vector Machine

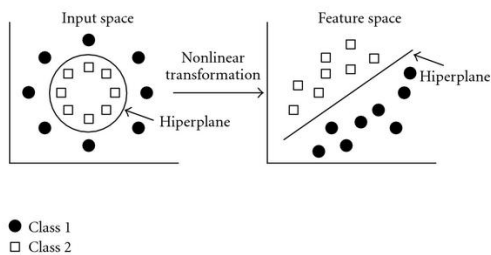
SVM classification [28–30] is rooted in statistical learning theory and pattern classifiers; it uses a training set,  $S = \{(x_i, y_i) : i = 1, \dots, m\}$ , where  $x_i$  is the characteristic vector in  $\mathbb{R}^n$ ,  $y_i \in \{1, 2\}$  represents the class, in this case 1 for open eyes and 2 for closed eyes, and  $m$  is the number of elements of  $S$ . From a training set, a hyperplane is built that permits classification between two different classes and minimizes the empirical risk function [30].

Mathematically, SVM consists in finding the best solution to the following optimization problem:

$$\begin{aligned} \text{min} \quad & f(\alpha) = \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m \\ & \gamma^T \alpha = 0, \end{aligned} \tag{12}$$

where  $e$  is an  $m$  by 1 vector,  $C$  is an upper bound,  $Q$  is an  $m$  by  $m$  matrix with  $Q_{ij} = y_i y_j k(x_i, x_j)$ , and  $k(x_i, x_j)$  is the kernel function. By solving the above quadratic programming problem, the SVM tries to maximize the margin between the data points in the two classes and to minimize the training errors simultaneously. Figure 17 depicts input space mapping to a high-dimensional feature space through a nonlinear transformation and its maximization process.

Figure 17 SVM representation.



#### 4.4.2. Eye Characteristic Extraction Using a Gabor Filter

The Gabor filter was used by Daugman for image analysis, changing the orientation, and scale [18] where these are multiscale and multiorientation kernels. They can be defined by the complex function

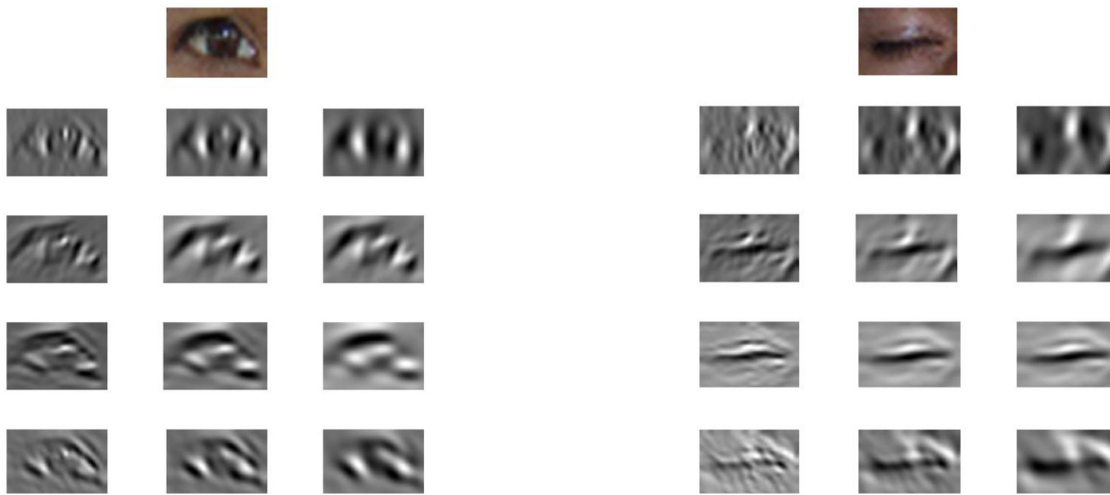
$$\begin{aligned} g(x, y, \theta, \phi) &= \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right) \exp(j2\pi\theta(x \cos(\phi) + y \sin(\phi))), \end{aligned} \tag{13}$$

where  $\theta$  and  $\phi$  are the scale and orientation parameters and  $\sigma$  is the standard deviation of the Gaussian kernel which depends on the spatial frequency to be measured, that is,  $\theta$ . The response of the Gabor filter to an image is obtained from a 2D convolution operation. Letting  $I(x, y)$  represent the image and  $G(x, y, \theta, \phi)$  denote the response of a Gabor filter with scale  $\theta$  and orientation  $\phi$  to an image at point  $(x, y)$  on the image plane by  $G(\cdot)$  is obtained using

$$G(x, y, \theta, \phi) = I(p, q) g(x - p, y - q, \theta, \phi) dp dq. \tag{14}$$

Various combinations of scales and orientations are more robust for the classification between open and closed eyes. Three scales and four orientations have been used to generate Figure 18; these are  $\{1, 2, 3\}$  and  $\{0, \pi/4, \pi/2, 3\pi/4\}$  which were obtained experimentally over an image size 30 by 20.

Figure 18 Gabor filter for [MediaObjects/13634\_2009\_Article\_2794\_IEq94\_HTML.gif] and [MediaObjects/13634\_2009\_Article\_2794\_IEq95\_HTML.gif]

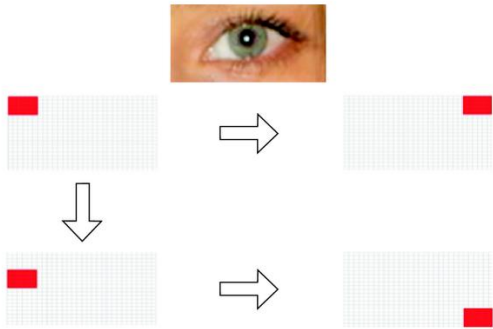


Once the response of the Gabor filter is obtained, the eye characteristic vector is extracted using a subwindow procedure described by Chen and Kubo [31] and denoted by  $d \in R^{360}$ . This vector is computed using (15) for each subwindow of size 5 by 6. Figure 19 shows the subwindow diagram

$$d_i^{\theta, \phi} = \frac{1}{30} \sum_{y=1:5} \sum_{x=1:6} G(x, y, \theta, \phi) \quad i = 1, \dots, 20.$$

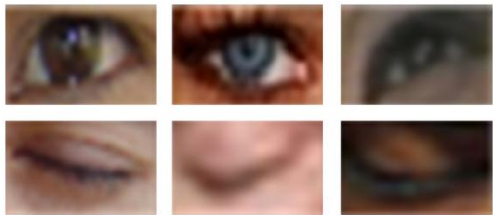
(15)

Figure 19 Subwindow images from the Gabor filter.



To perform this analysis a training set has been built which consists of both open and closed eyes. The images come from diverse sources, under different illumination conditions and are from different races. A further important aspect of this eye database is that it contains images of different eye colors, that is, blue, black, and green. Figure 20 shows several examples of this database.

Figure 20 Examples from the eye database for daylight illumination.



Previous to SVM training, it is crucial to preprocess each image where this procedure involves histogram equalization, filtering using a median filter, followed by the sharpen filter. The median filter is used to reduce image noise, and the sharpen filter enhances the borders.

The main objective of the SVM training is to obtain the best parameters and the best kernel that minimizes (5). After several SVM training experiments, it was decided to use the RBF kernel, that is,  $k(x_i, x_j)$  is  $\exp(-\gamma \|x_i - x_j\|^2)$ ,  $C = 30$ , and  $\gamma = 0.0128$ , where these parameters achieve a high training classification rate

of close to 93%.

Table 3 presents results using this method computed for several sequences of drivers. It demonstrates a high correct rate of classifications.

Table 3 Result of eye state analysis.

Driver	Total frames	Eyes open	Eyes closed	Correct rate
D1	960	690/700	258/260	98.90%
D2	900	520/560	339/340	96.27%
D3	500	388/400	99/100	98.00%
D4	330	150/170	152/160	91.61%
D5	1400	891/980	401/420	93.19%

## 4.5. Drowsiness Index

The eye-blinking frequency is an indicator that allows the level of driver drowsiness (fatigue) to be measured. As in the works of Horng et al. [12] and Dong and Wu [3], if for five consecutive frames or during 0.25 seconds the eye is identified as being closed the system issues an alarm cue, PERCLOS [7], which is also implemented in this system.

Figure 21 presents an instantaneous result for this system on a driver's image whereas Figure 22 pictures the drowsiness evolution index graph for a driver's drowsiness sequence. Additionally, other examples are shown in Figure 25.

Figure 21 Day system instantaneous result.

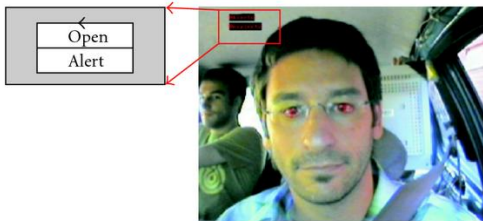
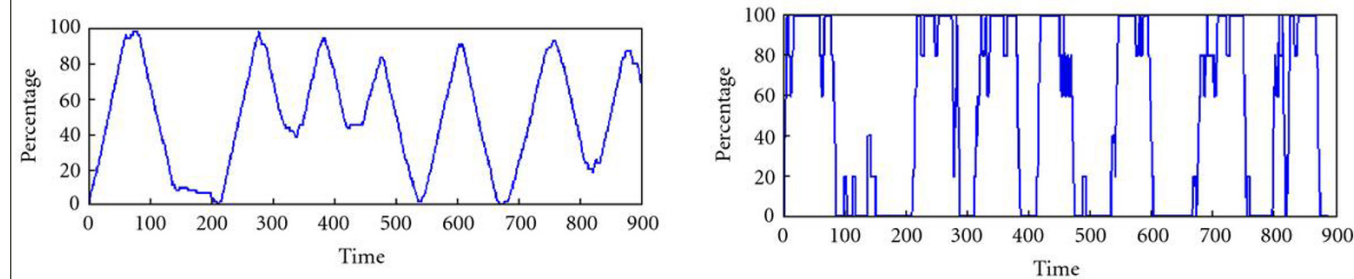


Figure 22. Drowsiness index graph for a 900-frame sequence of a drowsy driver: (a) Perclos, (b) Horng-Dong and Wu index.



## 4.6. Distraction

It is estimated that 20% of traffic accidents are caused by driver distraction [1]. To detect this characteristic the driver's face should be studied because the pose of the face contains information about one's attention, gaze, and level of fatigue [4]. To verify driver distraction the following procedure has been implemented.

### 4.6.1. Face Orientation

The driver's face orientation is estimated using the eye position, with

$$\theta = \tan^{-1}\left(\frac{\Delta x}{\Delta y}\right) \tag{16}$$

where  $\Delta x = x_2 - x_1$ ,  $\Delta y = y_2 - y_1$ ,  $(x_1, y_1)$ , and  $(x_2, y_2)$  correspond to the left and right eye positions, respectively. Equation (17) provides the classification limits. Figure 23 depicts an example of the face orientation and in Figure 24 an extended example is shown for the driver's face orientation using information from the eyes for a sequence of images

$$\begin{aligned} \text{Left} & \text{ if } \theta > 8^\circ, \\ \text{Front} & \text{ if } |\theta| \leq 8^\circ, \\ \text{Right} & \text{ if } \theta < -8^\circ. \end{aligned} \tag{17}$$

Figure 23 Face orientation.

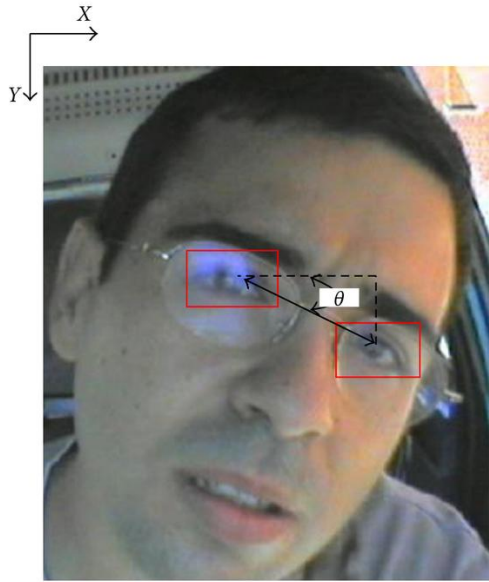


Figure 24 Head-orientation monitoring over time in a 100-frame sequence.

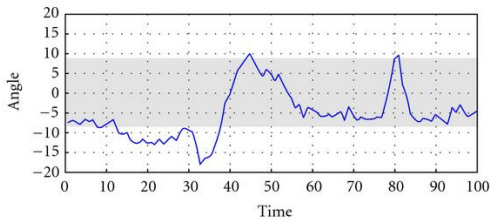
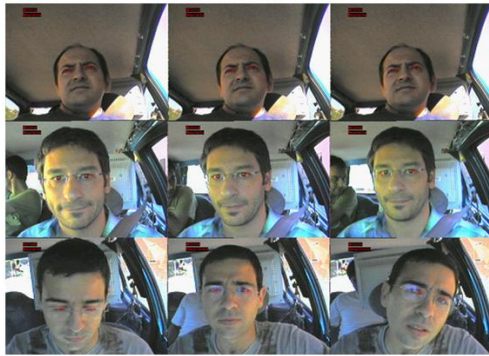


Figure 25 Different stages of the proposed algorithm during daytime conditions and for several time instants, driving conditions and different drivers.



#### 4.6.2. Head Tilt

The method described above presents problems when a monocular camera is used, and so, to overcome this drawback, this contribution has implemented a head-tilt based on neural networks. Keeping in mind that the driver face database is composed of face examples for five different orientations, the face is passed to the neural network to determine its orientation, specifically for the up and down cases. If the system detects that the face position is not looking straight on, an alarm cue is issued to alert the driver of a danger situation.

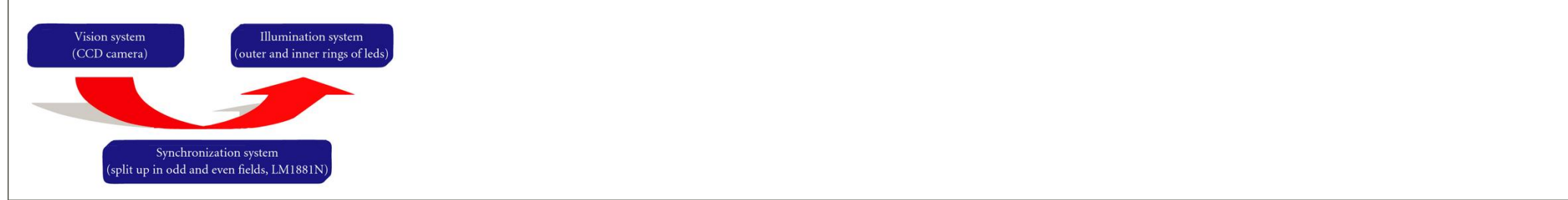
## 5. Night System Design

In this part of the work, the night system will be described, where this is based on the algorithm scheme shown in Figure 1(b). Note that it is composed of both software and hardware platforms. The main difference between this and the previous system is in perception system.

### 5.1. Perception System

The perception system is used to acquire visual information of the driver's head during nocturnal driving. It consists of three parts: a vision system, a synchronization system, and an illumination system; see Figure 26.

Figure 26 Perception system schema.



The first part is composed of a miniature CCD camera where the IR filter has been removed. This camera generates a composite video signal (PAL, phase alternating line). The second part takes the signal from the camera and splits it up into odd and even fields using the LM1881N video sync separator. In the third and last part, the illumination system, based on near-infrared (NIR) light (700–900 nm), contains two sets of NIR light-emitting diode (LED) rings (inner and outer) which produce odd and even image fields where the pupil effect is highlighted [3, 4]. The inner led ring surrounds the camera, while the outer leds are placed over a ruler in a symmetric position around the camera. Figure 27(a) shows this system, and Figure 27(b) provides an example of an illuminated driver's face from the NIR illuminator.

Figure 27 Perception system using infrared illumination (a) vision system, (b) drivers image, and (c) odd (top) and even (bottom) image fields.



Each frame is deinterlaced in both odd and even fields which contain the dark and bright pupil images, separately. Hence, the height of the odd and even image fields is a medium of the original image; this procedure can be seen in Figure 27(c): the top photograph is the even image field, and the bottom is the odd image field. Even an odd image will be used later on for eye detection.

### 5.2. Eye Detection

The bright effect pupil is the main principle behind locating the position of the eye. To do this sense, three images have been generated from the initial driver image, these are the difference image ( $I_D$ ), the edge image ( $I_E$ ), and the bright part of the fast radial symmetry transform (FRST) image [32] ( $I_F$ ).

The first image is computed using the absolute value of the pixel difference between the even and odd image fields; see (18). In this image most of the background and external light illumination has been removed, and the pupils appear as the brightest part

$$I_D(x, y) = |I_E(x, y) - I_O(x, y)|.$$

The second and third images are obtained using the Sobel Operator and the bright part of the FRST [32] over



the difference image, respectively; these may be seen in Figures 28(a), 28(b), and 28(c)

$$I_G = \sqrt{\|I^* S_x\|^2 + \|I^* S_y\|^2},$$

where  $S_x$  and  $S_y$  are given by (5).

Figure 28 Difference (a), edge (b), and FRST (c) and their binary images (d), (e) and (f), respectively.



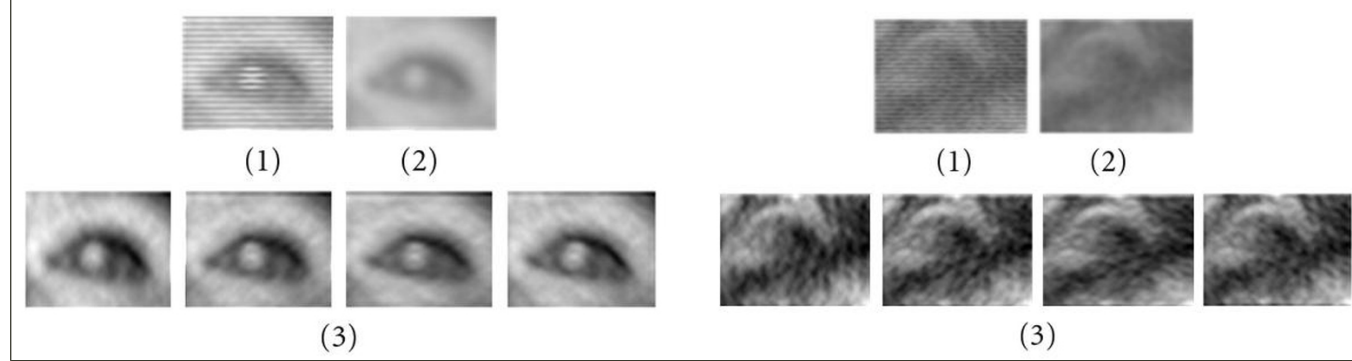
Most researchers only make use of the difference image for pupil detection; however, in real driving conditions, this image deteriorates due to external illumination, vibrations, and so forth and is also very sensitive to lighting conditions. In such circumstances, it is necessary to incorporate more robust information to improve the detection step. Therefore, in this paper, the edge and FRST images have been implemented to obtain enhanced results considering the aforementioned drawbacks.

Once all the images used to detect the eyes have been specified, the next step is to compute a binary threshold for the difference, edge, and FRST images. In the first of these, the threshold is obtained from a systematic analysis of its histogram, where two groups are formed. In the second case, the histogram is modelled using a Gamma distribution function where the 90% cumulative interval provides the threshold. Finally, in the third image, the maximum histogram value produces the required threshold level. This yields three binary images consisting of binary blobs that may contain a pupil.

The pupils are detected by searching within the entire image for the location of two blobs that satisfy a particular size, shape, and distance constraints. To remove false blobs an unsupervised classifier has been implemented, in this case, the SVM classifier, which is based on statistical learning theory and pattern classifiers. It uses a training set,  $S = \{(x_i, y_i) : i = 1, \dots, m\}$ , where  $x_i$  is the characteristic vector in  $R^n$ ,  $y_i \in \{1, 2\}$  represents the class, in this case 1 for open eyes and 2 for noneyes, and  $m$  is the number of elements of  $S$  [28]. Before training the SVM, it is crucial to preprocess each image, where this procedure involves histogram equalization, filtering using the median filter, followed by the sharpen filter. The median filter is used to reduce noise in the image, and the sharpen filter is used to enhance the borders. After this procedure a Gabor filter [31] is computed on each image of the eye database. Some combinations of scales and orientations are more robust for the classification between eyes and noneyes. One scale and four orientations have been used in this research; these are  $\{0\}$  and  $\{0, n/4, n/2, 3n/4\}$  which have

been obtained experimentally from an image size of 30 by 20 (in pixels). To perform this task, a training set has been built which consists of open eyes and noneyes, where an example is shown in Figure 29. Once the response of the Gabor filter is obtained, the eye characteristic vector is extracted and denoted by  $d \in \mathbb{R}^{600}$ .

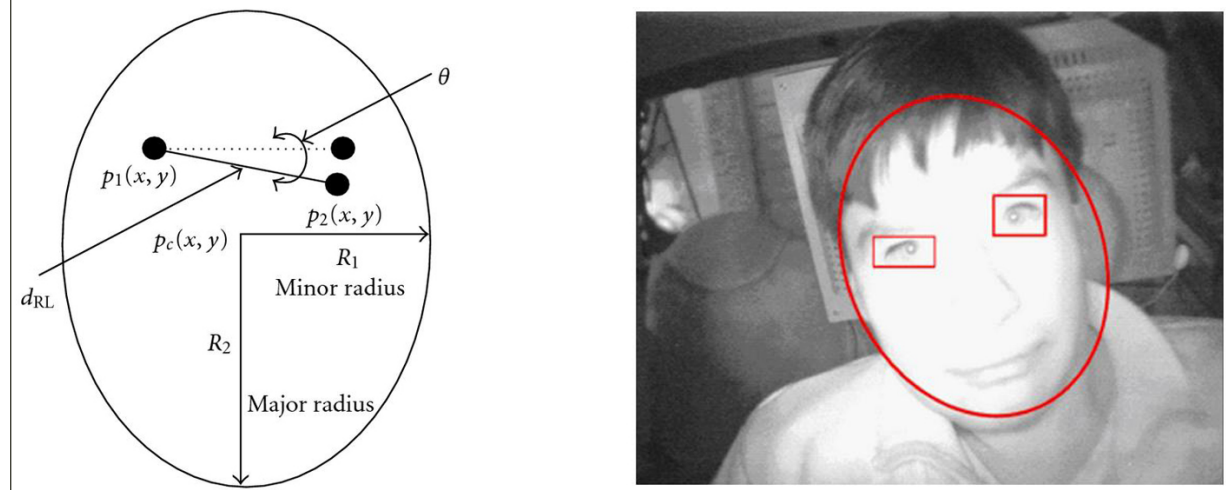
Figure 29 Gabor filter response for [MediaObjects/13634\_2009\_Article\_2794\_Ibq115\_HTML.gif] and [MediaObjects/13634\_2009\_Article\_2794\_Ibq116\_HTML.gif]



After several SVM training experiments, it was decided to use the RBF kernel, that is,  $K(x_i, x_j)$  is  $\exp(-\gamma \|x_i - x_j\|^2)$ ,  $C = 100$ , and  $\gamma = 0.128$ ; these parameters achieve a high training classification rate of about 93%.

This exhaustive search detects the pair of eyes; this is followed by an ellipse fitting which is applied to each pupil, and the center of the resulting ellipse is the position of the detected pupil. This process is presented in Figure 30(b). These will be used later to initialize the eye tracker.

Figure 30 (a) Human face model using the eye position, (b) face and eye detection.



### 5.3. Face Detection

Once the eyes have been located, the system continues with driver face detection. To perform this task, a human face model has been developed considering the pupil's position and face anthropometric properties [19]. Let  $p_1(x, y)$  and  $p_2(x, y)$  be the center position of the right and left eye, respectively, and  $d_{RL}$  their distance (in pixels); the area of the face is obtained from the following equations:

$$d_{RL} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

$$\theta = \tan^{-1} \left( \frac{y_2 - y_1}{x_2 - x_1} \right),$$

$$R_1 = 1.5 d_{RL},$$

$$R_2 = d_{RL},$$

$$p_c(x, y) = (x_1 + 0.5 d_{RL}, y_1 + 0.3 d_{RL}).$$

where  $p_c(x, y)$  is the centre of the face, and  $R_1$  and  $R_2$  are the axes of the face ellipse. Figure 30 depicts this model and its result.

### 5.4. Tracking

The tracking process has been developed using the Condensation algorithm for face and eye tracking.

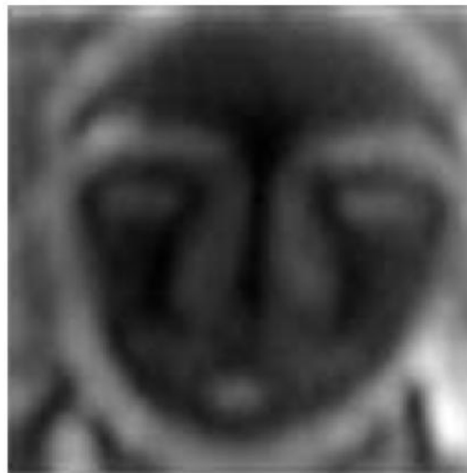
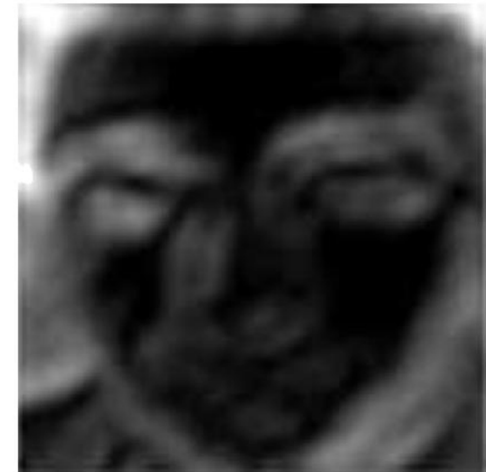
### 5.4.1. Face Tracking

This system uses the neural networks [33] and models and parameters proposed previously for face tracking. Now that the elements for tracking have been identified, only the backpropagation neural network training is developed in this section. Before training, a preprocessing step that consists of using a Gabor filter [18] with one scale and two orientations has been implemented. After this, the characteristic vector which consists of gray-level values of pixels coming from the face image is extracted. The rate of classification subsequent to the training is more than 92%. Figure 31 shows several face examples, and Table 4 presents experimental results. In this table, the true position has been retrieved manually.

**Table 4** Results of face and eye tracking and eye state analysis.

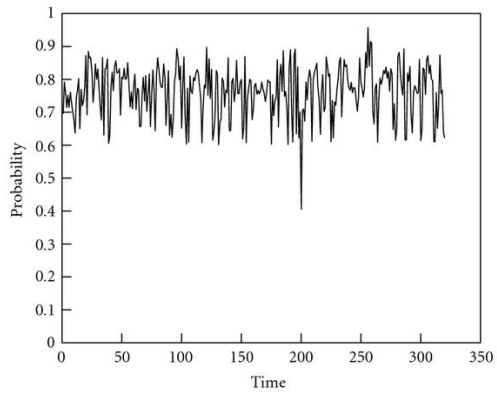
Driver	Total frames	Face tracking		Eye tracking		Eye state		
		Tracking failure	Correct rate	Tracking failure	Correct rate	Eyes open	Eyes closed	Correct rate
D1	800	40	95.00%	14	98.25%	690/700	97/100	97.78%
D2	646	29	95.51%	29	95.51%	500/530	100/116	90.27%
D3	600	20	96.67%	24	96.00%	345/244	206/226	91.69%

**Figure 31** Examples of the face database and its Gabor filter response.



The density function of the initial state is  $P(x_0) = N(x_0, \Sigma_0)$ , where  $\Sigma_0$  is computed using the previous face detection method, and  $\Sigma_0$  is given in [4]. Figure 32 shows the a posteriori density function of the face center tracking.

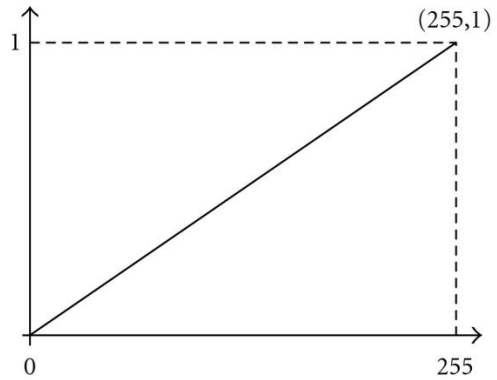
Figure 32 Estimated value of the a posteriori density of the face-center in a 350-frame sequence using the proposed tracker; the face is detected in the fourth frame.



### 5.4.2. Eye Tracking

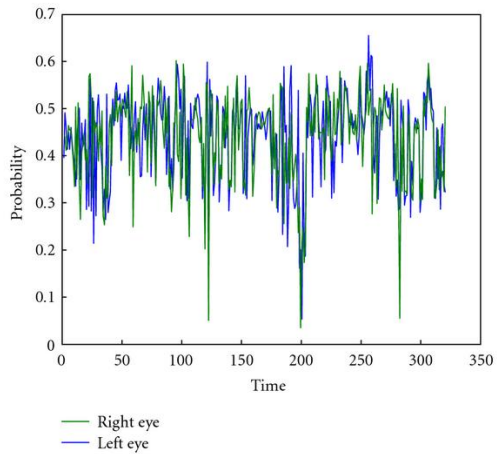
For this task, the state of the eye is characterized by its position and velocity. These parameters are also described in the diurnal system. To evaluate the probability observation density, a triangular density function based on the value of the difference image has been used (Figure 33). This function takes into account the gray-level value for the intensity of the illumination system.

Figure 33 Triangular density function.



CA is initialized when the eyes are detected from the method described in the previous section plus a white noise. In Table 4 the eye tracking results are presented, which have been carried out in several sequences of images. Again, the true position has been retrieved manually. Additionally, Figure 34 shows the a posteriori density function of both eyes.

Figure 34 Estimated value of the a posteriori density of the eye center in a 350-frame for right and left eyes, the eyes are detected in the fourth frame.



To evaluate the probability observation density, a triangular density function based on a value from the difference image has been used. CA is initialized when the eyes are detected using the method described in the previous section plus a white noise. Table 4 shows the eye tracking results which have been obtained from several image sequences.

## 5.5. Eye State Detection and Drowsiness Index

To identify drowsiness from an eye analysis, knowledge of the eye's state is required, that is, open or closed, in time and to develop an analysis over large periods of time, that is, to measure the time spent in each state. Classification of the open and closed state is complex due to changes in the shape of the eye, the changing position, and face rotations, as well as variations in twinkling and illumination, and so forth. All of these factors make it difficult to reliably analyze the eyes. However, when using the edge and FRST images, the eye state may be computed satisfactorily.

The PERCLOS [7] has been implemented in this system. Figure 35 presents an instantaneous result of this system obtained from a driver's image, and in Figure 36 the evolution of the drowsiness index graph for a sequence of driver drowsiness is presented.

Figure 35 System instantaneous result.

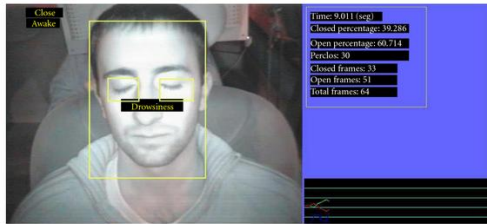
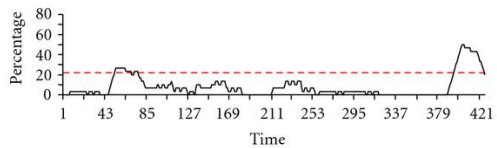


Figure 36 Drowsiness index graph in a 341-frame sequence of a drowsy driver.



## 5.6. Distraction

This method is similar to the previous case, once the face is continuously located in time: a neural network is used to determine its orientation and to verify the driver's level of distraction. If the system detects that the face position is not facing forward, an alarm cue is issued to alert the driver of a danger situation.

## 6. Conclusions

In this paper, a research project to develop a nonintrusive and autonomous driver drowsiness system based on Computer Vision and Artificial Intelligence has been presented. This system uses advanced technologies which analyze and monitor the state of the driver's eye in real-time and for real driving conditions; this is

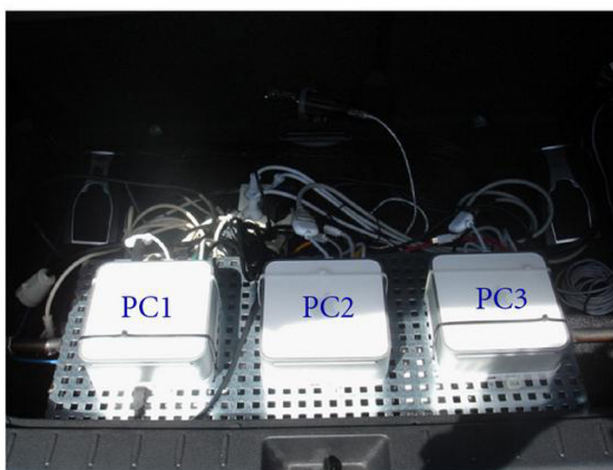
driving conditions for both daytime and nocturnal situations.

In the first case, based on the results presented in Tables 1, 2, and 3, the algorithm proposed for eye detection, face tracking, and eye tracking is shown to be robust and accurate for varying light, external illumination interference, vibrations, changing backgrounds, and facial orientations. In the second case, and as presented in the results of Table 4, the system is also observed to provide agreeable results.

To acquire the data required to develop and test the algorithms presented in this paper, several drivers have been recruited and were exposed to a wide variety of difficult situations commonly encountered on roadways, for both daytime and nocturnal conditions. This guarantees and confirms that the experiments presented here are proven to be robust and efficient for real traffic scenarios. The images were taken using two cameras within the IVVI (Intelligent Vehicle based on Visual Information) vehicle (Figure 27(a)): a pin-hole analog camera connected to a frame-grabber for the nocturnal illumination and a fire-wire camera for the diurnal use. Besides that, the hardware processes 4-5 frames per second using an Intel Pentium D, with 3.2 GHz, 2 GB, RAM memory and MS Windows XP.

IVVI is an experimental platform used to develop the driver assistance systems for real-life driving conditions. The most up to date version is a Nissan-Note car; see Figure 37(a). This vehicle is equipped with a processing system, which processes the information coming from the cameras. The processing system is composed of three personal computers (Figure 37(b)).

Figure 37 (a) IVVI vehicle, (b) processing system.



For future work, the objective will be to reduce the percentage error, that is, reduce the amount of false alarms; to achieve this, additional experiments will be developed, using additional drivers and incorporating new analysis modules, for example, facial expressions.

BodyRef	
FileRef	: BodyRef/PDF/13634_2009_Article_2794.pdf
TargetType	: OnlinePDF

## Acknowledgments

This paper was supported in part by the Spanish Government through the CICYT projects VISVIA (Grant TRA2007-67786-C02-02) and POCIMA (Grant TRA2007-67374-C02-01).

## References

1. Brandt T, Stemmer R, Mertsching B, Rakotonirainy A: **Affordable visual driver monitoring system for fatigue and monotony.** *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04), October 2004* 7: 6451-6456.
2. Tian Z, Qin H: **Real-time driver's eye state detection.** *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety, October 2005* 285-289.
3. Dong W, Wu X: **Driver fatigue detection based on the distance of eyelid.** *Proceedings of the IEEE International Workshop on VLSI Design and Video Technology (IWVDVT '05), May 2005, Suzhou-China* 397-400.
4. Ji Q, Yang X: **Real-time eye, gaze, and face pose tracking for monitoring driver vigilance.** *Real-Time Imaging* 2002,8(5):357-377. 10.1006/rtim.2002.0279
5. Bergasa LM, Nuevo J, Sotelo MA, Vázquez M: **Real-time system for monitoring driver vigilance.** *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2004* 78-83.

6. Fletcher L, Petersson L, Zelinsky A: **Driver assistance systems based on vision in and out of vehicles.** *Proceedings of the IEEE Symposium on Intelligent Vehicles, 2003* 322-327.
7. NHTSA : *Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management.* National Highway Traffic Safety Administration, Washington, DC, USA; 1998.
8. Hagenmeyer L: *Development of a multimodal, universal human-machine-interface for hypovigilance-management-systems, Ph.D. thesis.* University of Stuttgart, Stuttgart, Germany; 2007.
9. Longhurst G: *Understanding Driver Visual Behaviour.* Seeing Machine, Canberra, Australia;
10. Armingol JM, de la Escalera A, Hilario C, Collado JM, Carrasco JP, Flores MJ, Pastor JM, Rodríguez FJ: **IVVI: intelligent vehicle based on visual information.** *Robotics and Autonomous Systems* 2007,**55**(12):904-916. 10.1016/j.robot.2007.09.004
11. D'Orazio T, Leo M, Distanto A: **Eye detection in faces images for a driver vigilante system.** In *Proceedings of the IEEE Intelligent Vehicles Symposium, June 2004, Parma, Italy.* University of Parma; 14-17.
12. Horng W-B, Chen C-Y, Chang Y, Fan C-H: **Driver fatigue detection based on eye tracking and dynamic template matching.** *Proceedings of the IEEE International Conference on Networking, Sensing and Control, March 2004* 7-12.
13. Viola P, Jones M: **Rapid object detection using a boosted cascade of simple features.** *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, December 2001* 511-518.
14. Branzan A, Widsten B, Wang T, Lan J, Mah J: **A computer vision-based system for real-time detection of sleep onset in fatigued drivers.** *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '08), June 2008* 25-30.
15. Ji Q, Zhu Z, Lan P: **Real-time nonintrusive monitoring and prediction of driver fatigue.** *IEEE Transactions on Vehicular Technology* 2004,**53**(4):1052-1068. 10.1109/TVT.2004.830974
16. Brooksheer JG: *Theory of Computation: Formal Languages, Automata and Complexity.* Addison Wesley Iberoamericana; 1993.
17. Grace R: **Drowsy driver monitor and warning system.** *Proceedings of Driving Assessment International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design, 2001*
18. Daugman JG: **Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters.** *Journal of the Optical Society of America* 1985,**2**(7):1160-1169. 10.1364/JOSAA.2.001160
19. Gejgus P, Sparka M: *Face Tracking in Color Video Sequences.* The Association for Computing Machinery; 2003.
20. Otsu N: **A threshold selection method from gray-level histograms.** *IEEE Transactions on Systems, Man and Cybernetics* 1979, **9**: 62-66.
21. Jafar I, Ying H: **A new method for image contrast enhancement based on automatic specification of local histograms.** *International Journal of Computer Science and Network Security* 2007.,**7**(7):
22. Wu Y, Liu H, Zha H: **A new method of detecting human eyelids based on deformable templates.** *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '04), October 2004* 604-609.
23. McLachlan GJ: *The EM Algorithm and Extensions.* John Wiley & Sons, New York, NY, USA; 1997.
24. Isard M, Blake A: **Condensation: conditional density propagation for visual tracking.** *International Journal of Computer Vision* 1998,**29**(1):5-28. 10.1023/A:1008078328650
25. Isard MA: *Visual motion analysis by probabilistic propagation of conditional density, Ph.D. thesis.* Oxford University, Oxford, UK; 1998.
26. Parker JR: *Practical Computer Vision Using C.* John Wiley & Sons, New York, NY, USA; 1994.
27. Satake J, Shakunaga T: **Multiple target tracking by appearance-based condensation tracker using structure information.** *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04), August 2004* **3**: 294-297.
28. Cristianini N, Shawe-Taylor J: *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge, UK; 2000.
29. Chang C, Lin C: **LIBSVM: a library for support vector machine.** 2001, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
30. Guyon I, Gunn S, Nikravesh M, Zadeh LA: *Feature Extraction: Foundations and Applications.* Springer, Berlin, Germany; 2006.
31. Chen Y-W, Kubo K: **A robust eye detection and tracking technique using gabor filters.** *Proceedings of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '07), November 2007* **1**: 109-112.
32. Loy G, Zelinsky A: **Fast radial symmetry for detecting points of interest.** *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2003,**25**(8):959-973. 10.1109/TPAMI.2003.1217601
33. Looney CG: *Pattern Recognition Using Neural Networks, Theory and Algorithms for Engineers and Scientists.* Oxford University Press, Oxford, UK; 1997.