International Journal of Control, Automation, and Systems (2012) 10(2):317-327 ISSN:1598-6446 eISSN:2005-4092
DOI 10.1007/s12555-012-0212-0 http://www.springer.com/12555

Driver's Eye Blinking Detection Using Novel Color and Texture Segmentation Algorithms

Artem A. Lenskiy and Jong-Soo Lee*

Abstract: In this paper we propose a system that measures eye blinking rate and eye closure duration. The system consists of skin-color segmentation, facial features segmentation, iris positioning and blink detection. The proposed skin-segmentation procedure is based on a neural network approximation of a RGB skin-color histogram. This method is robust and adaptive to any skin-color training set. The largest remaining skin-color region among skin-color segmentation results is further segmented into open/closed eyes, lips, nose, eyebrows, and the remaining facial regions using a novel texture segmentation algorithm. The segmentation algorithm classifies pixels according to the highest probability among the estimated facial feature class probability density functions (PDFs). The segmented eye regions are analyzed with the Circular Hough transform with the purpose of finding iris candidates. The finial iris position is selected according to the location of the maximum correlation value obtained from correlation with a predefined mask. The positions of irises and eye states are monitored through time to estimate eye blinking frequency and eye closure duration. The method of the driver drowsiness detection using these parameters is illustrated. The proposed system is tested on CCD and CMOS cameras under different environmental conditions and the experimental results show high system performance.

Keywords: Driver awareness control, facial features detection, iris detection and tracking, skin segmentation, texture segmentation.

1. INTRODUCTION

Falling asleep while driving, is a major cause of road accidents. Some of these accidents are the result of the driver's medical condition. However, a majority of these accidents are related to driver fatigue, drowsiness, and driver inattention caused by various distractions inside and outside the vehicle. Car accidents associated with driver fatigue are more likely to be serious, leading to serious injuries and deaths. According to [1] in southwest England sleep related vehicle accidents made up nearly 16 % in general, and over 20 % of midland motorways. The European Transport Safety Council [2] states that driver fatigue is conservatively estimated to be a factor in about 20 % of road crashes in Europe. The Royal Society for the prevention of accidents, [3] reviewed literature on the causes of care accidents in the USA, UK, Australia, Germany, New Zealand, Norway, and Israel. It found that accidents as a result of driver fatigue ranged from 5 to 25 % of all accidents, varying by country. Particularly, in the United States [4] an estimated 1.35 million drivers were involved in a drowsy driving related crash between 1998 and 2003.

Other than driver fatigue, causes of serious car accidents include activities such as toggling the car audio system, speaking on the phone, and text messaging. Text messaging on a cell phone is associated with the highest risk of all cell phone related tasks [5]. In case of heavy vehicle drivers the risk of crashing while texting on the phone is increased as much as 23.2 when compared to an undistracted driver.

This relatively unexplored area of driver safety and accident prevention caused by drowsy and inattentive drivers attracted the immense attention of psychologists, engineers, and specialists in the area of computer vision. One of the approaches to help solve this problem comes from the visual monitoring of a driver's awareness through tracking and analyzing facial features. In this paper we propose a complete eye monitoring system that detects eyes that have been closed for longer than a predefined threshold as well the system is capable of detecting eye frequency for the further driver's attention monitoring.

Each step in the proposed system is compared with existing state-of-the-art techniques in Section 2. The descriptions of the face detection algorithm and the facial feature extraction technique are presented in Section 4. Section 5 describes the iris localization and tracking algorithm and Section 7 presents the experimental results. Concluding remarks are giving in Section 8.

Manuscript received December 2, 2010; revised October 25, 2011; accepted December 6, 2011. Recommended by Editorial Board member Dong-Joong Kang under the direction of Editor Young-Hoon Joo.

This work was supported by the 2006 Research Fund of University of Ulsan.

Artem A. Lenskiy is with the School of Electrical, Electronics & Communication Engineering, Korea University of Technology and Education, Korea (e-mail: lensky@kut.ac.kr).

Jong-Soo Lee is with the School of Computer Engineering and Information Technology, University of Ulsan, Korea (e-mail: jsoolee@ulsan.ac.kr).

* Corresponding author.

2. THE SYSTEM OVERVIEW AND COMPARISON WITH EARLY PROPOSED ALGORITHMS

Numerous approaches have been proposed to detect driver irises. They can be divided into approaches that use cameras with IR illumination and those that use monocular color cameras. Cameras with IR illumination simplify the problem of iris detection. When an infrared LED is located at the camera axis, the irises appear as two bright spots caused by the reflection of the blood-rich retina. Thus, an IR illumination based approach gained high popularity in eye detection [6] and consequently driver attention monitoring tasks.

In [7,8] a system for monitoring driver vigilance is proposed. The main idea consists in placing two cameras at different angles. The camera axis coincides with two coplanar concentric rings where along their circumferences a number of IR LEDs are evenly and symmetrically distributed. One camera has a wide range view for head tracking, and the other has a narrow view focusing on eye detection. Hammoud *et al*. [9] proposed a complete driver drowsiness detection system that detects irises in the near infrared spectrum.

Although IR based approaches perform reasonably at night time, it was noted [4] that those methods often malfunctioned during daytime due to the presence of sunlight. Moreover, when a driver is not looking straight the pupils' reflection drops down, making them difficult to detect. Another disadvantage of IR based approaches is the necessity of installing an IR LEDs setup. In comparison with IR cameras, CMOS and CCD cameras are passive, meaning there is no IR radiation. The effect of long term IR radiation should be studied to guarantee that there is no danger to eye health [10]. CMOS cameras are relatively inexpensive and ergonomic. Furthermore, according to [4] 52 % of drivers nodded off while driving between the times of 6:00 a.m. and 9:00 p.m. comprising a day's majority of bright daylight.

As a consequence, using IR cameras during those hours is impractical since IR cameras are inefficient under direct sunlight. In the case of a color camera, there is a possibility to take color information into account for skin-color segmentation purposes [11]. The best solution lays in a combinational approach when a color CCD camera is used during daytime and IR LEDs are turned on when the brightness is below a threshold. In this paper we focus on iris tracking using color cameras that allows us to take into account skin-color information.

The car driver monitoring system consists of learning and performing stages (Fig. 1). At the former stage the system is learned from training image pairs to distinguish skin-colors and detect facial features. Then, this information is used during the iris and blinking detection process. The iris and blinking detection process itself consists of the following steps:

1) Skin color segmentation and face detection;
2) Segmentation of facial region into close/open eyes, mouth, nose, eye brows regions;
3) Iris detection in open eye regions;
4) Tracking of irises.

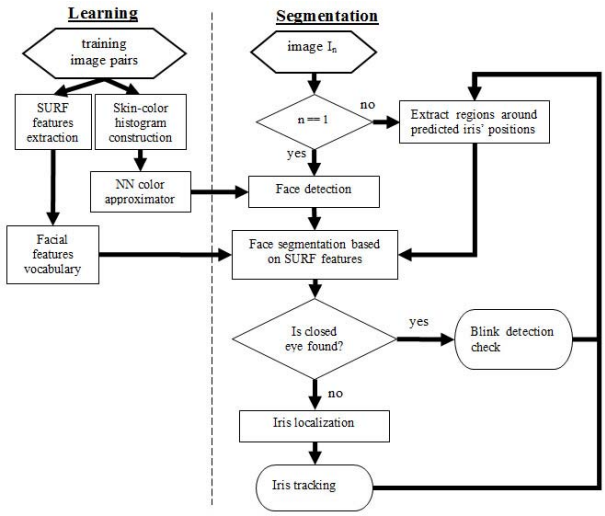


Fig. 1. Flowchart of the eyes' positions and states detection.

The overall block scheme of the proposed system is shown in Fig. 1.

2.1. Skin color segmentation and face detection

The skin-color segmentation process is commonly done in RGB, HSI or YCbCr color spaces. Some authors have heuristically found an RGB to 2-dimensional color space transform, then approximated the skin color domain in 2D space [12-16].These color spaces along with manual skin-domain approximation methods [11,17] are not capable of finding complex boundaries of skin-color domains. One approach to define complex boundaries of skin-color domains is to train an artificial neural network to separate skin and non-skin colors. Several applications of neural networks for skin color filtering have been proposed. A two layer multi-layer perceptron (MLP) with two inputs and three hidden neurons was applied by Chen *et al*. [18]. In their work the RGB color space was transformed into normalized CIE XYZ color space, then the values of X and Y coordinates served as inputs for MLP. Another attempt using the MLP skin-color filter was proposed by Seow [19], where one additional neural network was used at the learning stage to interpolate spatial skin region in each of the training images. The two hidden layers MLP then learns to distinguish skin-colors obtained from interpolated regions and non-skin colors from the rest of the image. Sahbi and Boujemaa [20] applied a two layer MLP with two hidden neurons and three inputs and outputs. This structure allowed them to extract principle skin color components. Lenskiy and Lee [21] applied one layer feed forward neural network to segment skin-color. They suggest an interesting and simple approach to train the network with negative samples uniformly distributed in the color space. As soon as the segmentation process is over, it is relatively easy to detect the face in the remaining skin-color segments using facial proportions.

Compared to the above described skin-color segmentation methods, we propose a skin-color segmentation algorithm based on the neural network trained to

approximate the skin-color histogram. This approximation takes into account the frequency of colors in the skin regions and assigns a small negative histogram value for colors in the non-skin regions. Using the histogram in our system, we could segment the skin colors of higher frequencies with more accuracy.

2.2. Facial features segmentation

Facial features approaches has been a hot research issue for a long time. Some approaches are based on raw images of facial features that are fed into a classifier, such as support vector machines [22] or neural networks [11,23,24]. Other authors prior to classification extract robust and compact features. The most popular features are based on wavelet transform [24,25], and particularly on Gabor transform [11,26] or secondary derivatives of the Gaussian kernel [27]. Some methods utilize geometrical properties of corresponding facial features. For instance, eye can be detected by calculating generalized projection function [28]. One more approach to localize eye regions is to consider the average intensity in the running window. Usually, the intensity of the eye region is lower compared to the rest of the face. This concept was applied in [29] and [30]. In the latter work the invariance to rotations is achieved with the use of Zernike Moments.

In our work we propose a new facial segmentation method that estimates probability density function using spatial locations of the salient points and their descriptors. SURF features are commonly used for 3D reconstruction and object recognition. Recently, SURF features were also proposed for texture segmentation [31,32]. Using SURF features we are able to obtain robust facial segmentation method as these features are invariant under brightness, rotation and scale changes.

2.3. Iris detection and tracking

For iris detection we use Circular Hough transform similarly to [33]. To further improve the performance we correlate each iris candidate with the predefined mask, and choose among all candidates only the one with maximum correlation value. To speed-up iris detection process and detect blinks we are separately tracking left and right eyes with two Extended Kalman filters. The iris positions and their states are stored for the purpose of drowsiness detection. The drowsiness detection is simply done by measuring the time of eye closure and checking if the eye has been closed longer than a predefined time threshold.

Briefly, the main contribution of this work can be summarized as follows:

- We propose a novel eye and lips detection algorithm based on SURF features and estimated conditional probability density functions;
- We propose a drowsiness detection system that utilizes driver's eye blink duration.

The robustness of the proposed algorithm relies on SURF features' invariance to projective transformations and on the assumption that SURF features of one class reside in close proximity to each other.

3. FACE DETECTION ALGORITHM

3.1. Training data preparation

The following sections present face and eye detection algorithms that are adaptive in terms of their ability to learn from training sets. Therefore, the detection quality strongly depends on the quality of the training set and the proper selection of training set is a task of high importance. We select a great number of training examples. This set contains images of people with varying skin-tones, under various lighting conditions, and with different facial expressions and head orientations. Depending on either skin-segmentation or facial features segmentation the post-processing procedures differ and is discussed in the following sections.

Each image in the selected training set is associated with a segmented hand map. The map is segmented into a predefined set of colors. The background is marked in black, as it has no interest and may vary significantly depending on vehicle interior design and environment.

Facial regions are marked with red, yellow, magenta, green, cyan and blue. More specifically lips are marked with red, eyebrows with yellow, nose with cyan, closed and opened eyes with magenta and green respectively and the rest of the face in blue. The blue region includes cheeks, forehead, and other skin-colored regions (Fig. 2). Pixels from this region are used in constructing skin-color model.

3.2. Skin color model

Several color spaces have been suggested, providing an easy method of separating skin-color domains. As mentioned previously, most of these methods apply heuristic approaches to customize boundaries of skin-color domains. We suggest a unified method based on an adjustable model obtained through the training procedure.

Our adaptive model utilizes a single layer perceptron (SLP) with sigmoid activation functions in hidden layer neurons and a linear activation function for the output neuron. The number of inputs corresponds to the number of color components in RGB space i.e., red, green, and blue. The number of neurons in the hidden layer is not restricted and can be customized according to the complexity of the training set. Using a trial and error method we set up a number of hidden neurons equal to 10.

The SLP is trained with a training set containing two subsets. The first subset is for positive, i.e., skin-color samples, and the second for negative non-skin color samples. We built a 3-dimensional histogram, where the

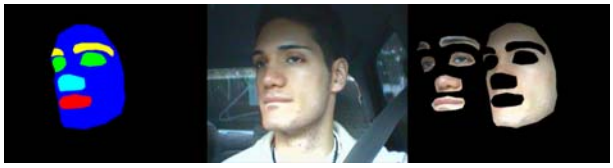


Fig. 2. Hand segmented map, a corresponding training image and extracted facial regions are shown on the left, center and right correspondingly.

intensity of each color component corresponds to a value on the orthogonal axis and the number of pixels with a particular color $\mathbf{x} = \{r, g, b\}$ appearing in all training images represents a histogram value $H(\mathbf{x})$ (Fig. 3). We are able to obtain the skin-color membership function by normalizing the histogram dividing it on the quantity representing the most often appearing color. In our case, the maximum value that the normalized RGB histogram takes equals 1, which is the maximum value that a sigmoid function in a hidden layer neuron can take. The target of the training procedure is to minimize the following criteria:

$$e(\mathbf{w}) = \sum_{\mathbf{x}} \left(f(\mathbf{w}, \mathbf{x}) - H(\mathbf{x}) \right)^2 \xrightarrow{\mathbf{w}} \min, \tag{1}$$

where f is a single layer feed forward neural network described as follows

$$f(\mathbf{w}, \mathbf{x}) = \sum_{i=1}^{10} w_i^{(2)} \sigma \left(\sum_{j=1}^{3} w_{i,j}^{(1)} x_j + w_0^{(1)} \right) + w_0^{(2)}, \tag{2}$$

where $\sigma(x) = \dfrac{2}{(1+e^{-2x})} - 1$ and $\mathbf{w} = \{w^{(1)}, w^{(2)}\}$ denotes matrices of interconnecting weights of input and output layers.

We reduce the original 256x256x256 RGB space to the size of 64x64x64 for two primary reasons. Firstly, a decrease in size leads to a fewer number of neurons needed for approximation of the skin-color domain which consequently leads to faster calculations. Secondly, even though our training set contains a great amount of skin-color pixels, there is still a chance that some skin-colors will not be within the training set and therefore by reducing dimensions we are able to reduce emptiness within the skin-color domain caused by the lack of appearance of particular skin-colors pixels. It is worthy to mention that the quality of skin segmentation based on reduced RGB space slightly reduces too. However, the experimental results show an acceptable segmentation quality.

The subset with negative samples represents colors from outside the skin-color domain. They are defined by uniformly positioning negative values (equals -0.01) around each positive sample, but avoiding positions with skin-colors (Fig. 3). Defining negative examples this way is reasonable, due to the dense skin color domain we obtained after RGB space compression.

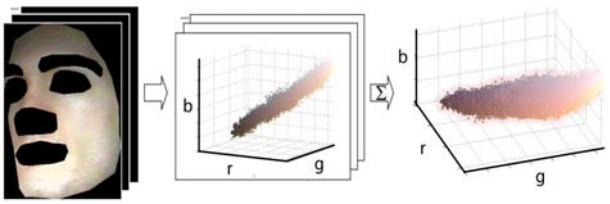


Fig. 3. Preparation of training samples resulting in cumulative histogram with negative samples (red dots).

3.3. Skin color model

The segmentation process is simply utilize the trained neural network to decide whether the pixel belongs to skin-colors or not. The decision rule Γ is set as

$$\Gamma(I(i,j)) = \begin{bmatrix} 1, & 0 < f(\mathbf{w}, I(i,j)) \\ 0, & f(\mathbf{w}, I(i,j)) \le 0, \end{bmatrix} \tag{3}$$

where $I(i, j)$ is an input image.

The results of skin-color segmentation contain a number of components. Many of them such as arms and clothes are leftover from the segmentation process and are subject to elimination. To identify which of the remaining components is the face, an opening operation is executed to fill in empty spaces caused by skin-color segmentation, and then connected components are extracted. The component with the largest number of pixels is considered as the facial region. The extracted facial region is then subject to the eye localization process.

3.4. Skin color segmentation

To speed up the segmentation algorithm, each image is decomposed into the image pyramid where the bottom layer is the original image and image at every higher layer is half the resolution of the image at the previous layer. To reduce the computational time we do not blur images with the low pass filtering as it is performed in the Gaussian pyramid. To decrease calculation, image resolution is reduced by simply decimating every second sample.

Then instead of processing each pixel in an input image, we start segmentation from the top layer of the pyramid. In the case of a four layer pyramid, the side of the top layer is eight times smaller than the side of the input image. Therefore, on average, the number of pixels is reduced by a factor of 64 and subsequently, so is the calculation time. For each layer in the pyramid a binary map is created. Initially the binary map is filled with zeros. Values with coordinates of pixels which are classified as skin-colors are filled with ones. The resulting binary map is interpolated to the next lower layer with double sides. Then, the pixels whose

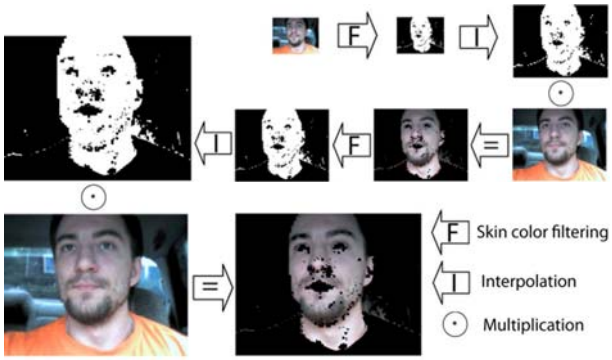


Fig. 4. Filtering starts at the lowest spatial resolution image (the top layer) and recursively back-propagated to lower layers by applying interpolation and overlapping interpolated images with the corresponding pyramid layers.

coordinates correspond to zeros in the binary map are removed in the lower layer of the pyramid. This process is repeated until the bottom layer is reached. Schematically two iterations of segmentation process are shown on Fig. 4.

4. FACIAL FEATURES EXTRACTION

4.1. Facial features model

In our approach, eyes are treated as textures. As a texture, each eye does not completely replicate the appearance of the other eye, although it looks very similar. We propose a supervised facial segmentation method based on the salient features classification. Firstly, a data set of training pairs is selected. Each pair contains an image with a face and hand segmented image, where correspondent facial regions are marked with different colors as was discussed above. Although, we are mainly concerned about eye group, we added other facial categories for a better separability in the feature space.

Each of the images is processed to extract speeded-up robust features (SURF). The original SURF features are comparably short descriptor vectors (64 dimensions) compared to SIFT feature (128). However, to speed-up the segmentation process instead of 64 dimensional features we operate with 36 dimensional features. Moreover, as it was shown in [34], 36 dimensional features are more suitable for texture segmentation. The SURF algorithm consists of three stages. In the first stage, interest points and their scales are localized by finding in scale-space the maxima of the determinant of the Fast-Hessian matrix. In the second stage, prior to computing the feature descriptor, the feature's orientation is determined. This stage calculates Haar wavelet responses for both x and y directions surrounding the interest point and the dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window. This direction is then used to create a rotated square around the interest point. Finally regularly sampled Haar responses within this window are combined per grid location to form the final descriptor [35]. In our system we experimented with the SURF algorithm as well as upright right version of the SURF (U-SURF). The latter one is a rotation dependent version of the SURF. It skips the second stage and as a result it is faster to compute. We found that the U-SURF shows better performance than the SURF.

Extracted features are arranged into six groups depending on their locations. Those features whose locations fell within the green or magenta segments are placed into the open eye or closed eye groups correspondingly. Those features which fell into the red segment are arranged into the lip group, while features from yellow regions are placed in the eye-brow group. The sixth class corresponds to the blue region and contains features which didn't fall in any of the above classes. These are features from other parts of the face. Features for correspondent regions from all training pairs

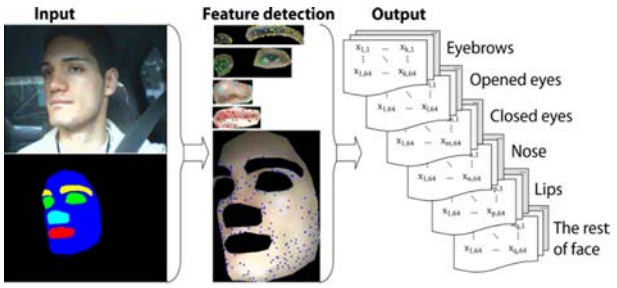


Fig. 5. Knowledge base construction: features are extracted and stored in the matrix form.

are concatenated into six matrices that constitute the facial features model (Fig. 5).

4.2. Facial features model

The next step after facial candidate is detected, involves facial regions segmentation using the facial features model. The segmentation process begins with extracting SURF features from the facial region detected earlier. The Euclidian distance is calculated between the descriptors of detected features and descriptors from facial features model. Then a probability of a SURF descriptor belonging to a class from the facial feature model is estimated. Therefore, for each extracted descriptor a six dimensional vector is associated. Each value in the vector corresponds to a probability of a SURF feature to belong to a corresponding class. Based on these values six probability density functions over image sized field is estimated using Parzen window method (Fig. 6(b)). Thus for any given pixel we are able to determine the probability that a pixel represents one of the facial features. We segment the facial region by selecting the class with the highest probability for each pixel. The segmentation process is performed pixel by pixel, therefore to speed-up the process, and input image is first subsampled to a lower resolution. The segmentation algorithm is summarized in the following list of steps:

1) Extract SURF feature (l_k, d_k) from a subsampled image I, where l_k, = (x_k, y_k) and d_k are features' locations and descriptors correspondingly.
2) For $q \in I$, select indexes of extracted features that reside within radius r around the pixel $q = (x, y)$:

$$T(q) = \left\{ j \middle| \left\| q - l_k \right\| < r, k = 1..N \right\}, \tag{4}$$

where N is the number of extracted SURF features.

3) Find minimal Euclidian distances v:

$$v_{k,i} = \begin{bmatrix} \min\limits_{j=1..N1} \left(\left\| d'_k - d^I_j \right\| \right) \\ \min\limits_{j=1..N2} \left(\left\| d'_k - d^{II}_j \right\| \right) \\ \cdots \\ \min\limits_{j=1..N6} \left(\left\| d'_k - d^{VI}_j \right\| \right) \end{bmatrix}, \tag{5}$$

where $k \in T(q)$, $i = 1..6$, d^I, d^{II}, …, d^{IV} – descriptors

from the corresponding vocabulary. *N1*, *N2*, ..., *N6* – numbers of descriptors in each class.

4) Apply the Parzen Window method to estimate the probability density functions(PDF) that pixel q belongs to the class c:

$$p_i(q) = \sum_{j=1}^{\#(T(q))} \exp\left(-\frac{v_{j,i}}{2\sigma_1^2} - \frac{\|q - l_k\|}{2\sigma_2^2} \right), \quad (6)$$

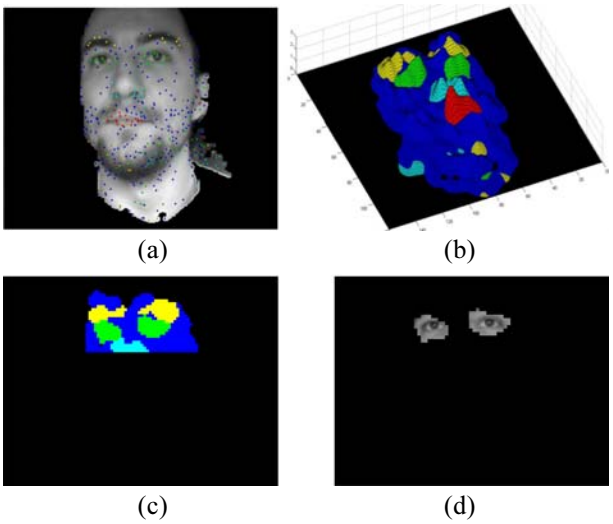
where i = 1..6 represents class index.

5) Each pixel $q = (x, y)$ is classified into one of the six classes by applying the following rule:

$$t(q) = \arg\max(p_i(q)). \quad (7)$$

If the greatest probability is less than a threshold, then a pixel's class is considered as unknown (marked black in the segmented image).

To increase the rate of correctly detected eye regions we reduce the search area by avoiding the part of the face below the nose (Fig. 6(c)). The number of pixels that represents the nose region must be higher than a predefined threshold. Usually the nose is easy to detect and in most cases it is clearly visible, regardless of presence of facial hair or glasses. We apply dilatation operation to the remaining regions. This guarantees that iris edges will be within the eye region. At this point two larges regions containing pixels form opened and closed eye classes are considered as detected eyes. If the region area is below a defined threshold, the region is omitted as there are no eyes present. If the majority of pixels in the combined opened and closed eye region belongs to either class then the whole region is classified as the one with the majority of pixels (Fig. 6(d)). Following this procedure we were able to detect blinks.



(a) (b) (c) (d)

Fig. 6. (a) the color and the intensity of features reflect their most probable class and its probability, (b) estimated probability density functions, (c) only segmented facial regions above the nose are shown, (d) detected eye regions.

5. IRIS LOCALIZATION AND TRACKING

5.1. Iris localization

In the case when an eye is detected as closed, the centriod of corresponding region is calculated and the coordinates are stored, otherwise the eye region is analyzed to localize the iris. In order to localize the iris, the circular Hough transform is applied [36]. The Hough transform operates on a binary image that we obtain through the Canny edge detector [37]. Due to the possible difference in eye size from person to person and slight variation in the distance between the camera and the driver, the Circular Hough transform is applied for a range of radii which is defined proportionally to the image size. The Hough transform generates an array called accumulator, where the indexes of the maximum value corresponds to positions of detected circles. Often there are positions in the array with equal maximum value. In this case all positions are stored for further analysis.

Next step is to select which of the radii and positions correspond to an iris and which are false. The solution to this task consists in generating a ring mask based on coordinates and radius of each of detected circles and calculating a normalized correlation between the mask and the eye image. The mask is a ring with an inner circle radius r filled with values equal -1.5 and the outer circle with radius $r_0 = r\sqrt{2}$ filled with 0.5. This choice of r_0 makes the area of inside circle equal to the area of the ring. The generated mask intends to simulate the phenomenon that the iris is always darker than the sclera, so normalized correlation coefficient k reflects how close the detected circle reassembles the eye appearance. The final criterion is a product of the following form (Fig. 7):

$$v = k\frac{a}{5.7r}, \quad (8)$$

where k is normalized correlation coefficient and a is the corresponding accumulator value, 5.7 is an approximation of 2π counted in pixel and suitable for r = [5..12]. It is used to normalize the length. In the case of complete

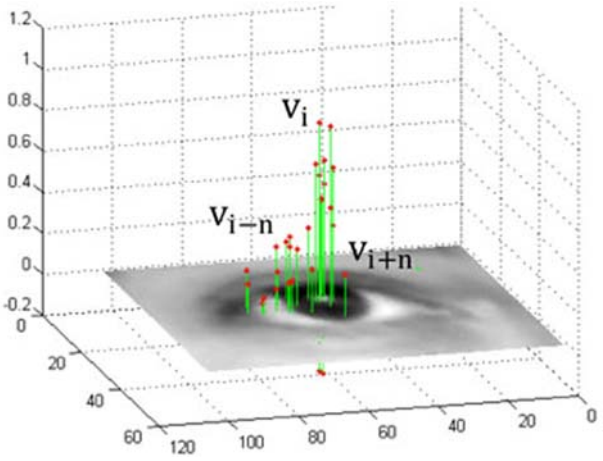


Fig. 7. The maxiumum of criteria (8) correposnds to the detected iris.

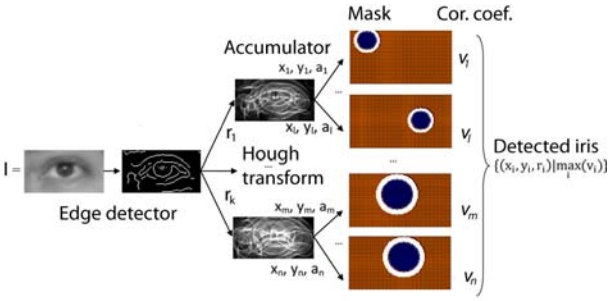


Fig. 8. The iris detection process.

circle, $\frac{a}{5.7r}$ approximately equals 1. The value of (8) is higher when edges are well visible and form a circle and also when internal circle area is dark and outer area is bright. The coordinates and radius corresponding to the maximum value of (8) are considered as the center and radius of the detected iris. Schematically the iris detection process is shown on Fig. 8.

5.2. Iris tracking

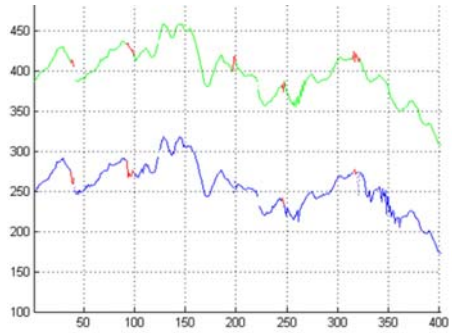
In this section we consider the task of iris tracking. It is uncertain whether the visible eye is left or right when only one eye is visible due to head rotations. Furthermore when the other eye is detected it is important not to interchange eye markers as being left or right. As a solution to this problem we suggest to track each eye separately using the Kalman filter in an alternate form [38]. The detected iris is classified as being left or right if its position is close to a predicted position by corresponding left or right eye tracking Kalman filter.

If one of the eyes is out of sight, the Kalman filter uses previously predicted values to estimate current position. This approach allows us to deal with situations when an iris for some reasons was not recognized.

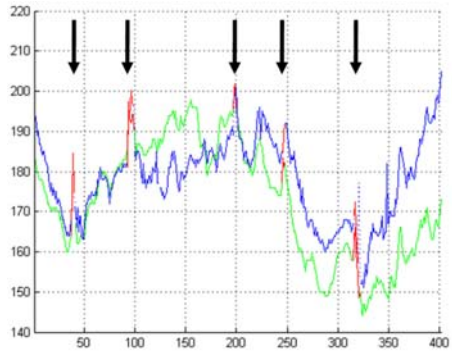
To reduce the computation time only regions around predicted by the Kalman filter iris positions are analyzed. The size of regions changes adaptively according to the covariance matrices used in the Kalman filters. In the case when one or both eyes were not detected, the whole image must be processed.

6. IRIS LOCALIZATION AND TRACKING

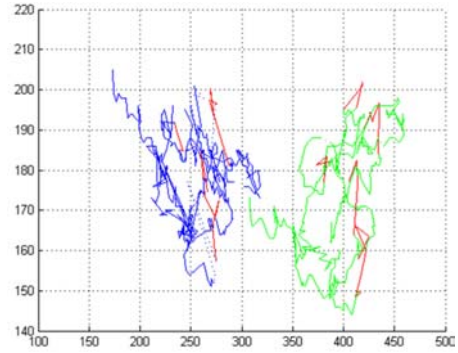
The presented system for eye movement monitoring allows us tracking in time positions of the eyes and their statuses. As we focused on the development of the eye movement monitoring system, we are not going into details on sophisticated algorithms for driver drowsiness detection and attention control, although we consider a straight-forward drowsiness detection algorithm since our system is capable of measuring such quantities as eye closure duration and eye blinking frequency. The drowsiness detection problem is a very difficult problem itself and information of blinking frequency and eye closure duration should be taken into account with caution. The difficulties are caused by multiple factors that are hard to control, yet they influence eye blinking frequency and duration. Such factors include environmental factors, burning of eyelids, bright illumination or eye drying air flows. Nevertheless, the recent research support the possibility of drowsiness detection based on the blink duration and frequency measurement. An interesting work has done by Caffier [39] demonstrated that the parameters blink duration and blink frequency change reliably with increasing drowsiness. Furthermore, the proportion of long closure duration blinks proved to be an informative parameter. Some techniques have been proposed for drowsiness detection which utilize such information as the percentage of time that the eyes are between 80 and 100 percent closed during a defined time interval [40] or such information as the eye-blinking frequency [41] and closure duration.



(a) Absciss is a time axis and ordinate is eye x-coordinate.



(b) Absciss is a time axis and ordinate is eye y-coordinate.



(c) Absciss and ordinate are x and y coordinates correspondingly.

Fig. 9. A 25 second sequence of iris tracking with detected five eye blinks.

Our drowsiness detection algorithm is based on the latter parameter. Even though the alert and drowsy conditions associated with dissimilar eye-blinking frequency and duration, their values are differs from person to person. As it was shown in [39] in 90 % of the people participated in their experiments, significant differences between both measurements were occurred. The mean value of blink duration in the alert condition case was 202.24 ± 6.07 ms and in the case of drowsy condition it was 258.57 ± 6.67 ms. The mean of the blinking frequency was 16.33 min^{-1} in the case of alert conditions compared to 15.83 min^{-1} in the case of drowsy conditions. We used eye closure duration to decide whether a driver is drowsy or not. So, if the eye has been detected as closed for longer than 220 ms, corresponding to 4 frames in the camera with the frame rate of 15 frames per second, then the system considers the situation as a driver being drowsy. This situation may happen in two cases either eyes are closed or the driver is nodding.

To check that our system can be used for drowsiness detection purpose we analyzed time series containing the eye states to find out intervals where both of the eyes were detected as closed for longer than 4 frames. Fig. 9 shows an example of detected eye blinking. The duration of each blink was within 4 frames interval and therefore no danger is detected. Examples of detected iris trajectories are shown on Figs. 10 and 11. Green and blue curves correspond to left and right iris positions. Red segments indicate that eyes are closed. Dashed segments indicate that only one eye is detected.

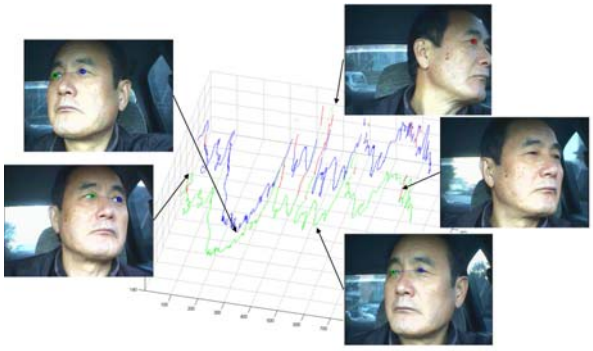


Fig. 10. Eyes trajectories and select video frames captured from a CCD camera are shown.

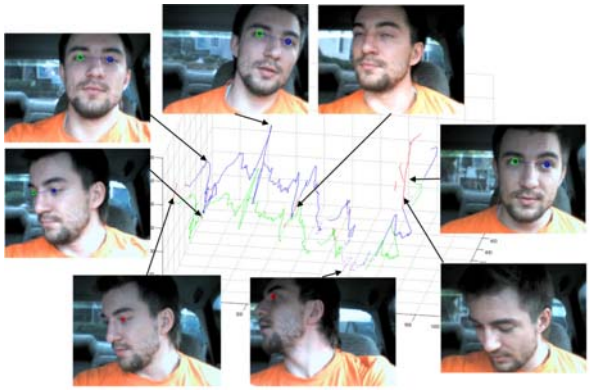


Fig. 11. Eyes trajectories and select video frames captured from a webcam are shown.

7. DISCUSSION

7.1. Results

We experimented with CCD and web cameras. The resolutions of CCD and web cameras were 768x576 and 640x480 correspondingly. The cameras were attached to the car's dashboard. We recorded 15 video sequences of around 250-400 frames each. For the training purpose we selected 120 images from 9 video sequences for each camera. Each video sequence was processed and then visually analyzed. By tracking eye movements separately, we could estimate detection rates for each eye. Tables 1 and 2 reports the percentages of correctly detected eyes for CCD and web cameras correspondingly. The number of frames in each video sequence is specified in the second column. The situation when the eye is visible and correctly localized is marked as true positive. True positive situations also include cases when the eye absence is recognized correctly. The case when the eye was incorrectly detected at the place where there is no eye is marked as false positive. If the eye is presented in the image but not detected the situation is recognized as false negative.

Although the resolution of images captured form the

Table 1. Detection rates with the CCD camere.

Sequence		Total number of frames	True positive	False positive	False negative	Success rate (%)
1	left	250	249	1	0	99.6
	right		247	0	3	98.8
2	left	250	245	1	4	98.0
	right		243	0	7	97.2
3	left	250	242	4	4	96.8
	right		244	3	3	97.6
4	left	257	236	9	5	91.8
	right		234	5	11	91.0
5	left	300	295	0	5	98.3
	right		300	0	0	100
6	left	300	286	0	14	95.3
	right		274	1	26	91.3
7	left	301	287	0	14	95.3
	right		294	0	7	97.6
8	left	300	291	2	7	97.0
	right		285	4	11	95.0
9	left	300	296	2	2	98.6
	right		287	5	8	95.6
10	left	316	302	8	6	95.5
	right		305	4	7	96.5
11	left	300	278	13	9	92.6
	right		298	0	2	99.3
12	left	300	296	0	4	98.6
	right		292	0	8	97.3
13	left	292	261	14	17	89.3
	right		270	4	18	92.4
14	left	300	266	6	28	88.6
	right		281	3	16	93.6
15	left	367	327	8	32	89.1
	right		336	4	27	91.5

Table 2. Detection rates with the CMOS camera.

Sequence		Total number of frames	True positive	False positive	False negative	Success rate (%)
1	left	330	321	5	4	97.2
	right		328	0	2	99.3
2	left	330	316	3	11	95.7
	right		308	7	15	93.3
3	left	340	303	13	24	89.1
	right		329	0	11	96.0
4	left	250	248	0	2	99.2
	right		243	1	6	97.2
5	left	250	239	2	9	95.6
	right		242	0	8	96.8
6	left	250	246	0	4	98.4
	right		243	1	6	97.2
7	left	300	299	1	0	99.6
	right		299	0	1	99.6
8	left	300	289	3	8	96.3
	right		299	1	0	99.6
9	left	247	224	19	4	90.6
	right		237	2	8	95.9
10	left	300	298	0	2	99.3
	right		279	4	17	93.0
11	left	300	259	14	27	86.3
	right		278	3	19	92.6
12	left	323	319	0	4	98.7
	right		312	0	11	96.6
13	left	330	325	1	4	98.4
	right		318	1	11	96.3
14	left	330	326	0	4	98.7
	right		321	2	7	97.2
15	left	340	334	0	6	98.2
	right		329	2	9	96.7

CCD camera is higher than images from the webcam, the average eye detection quality is slightly lower when the CCD camera is used (95.3 % with the standard deviation 3.4 %). In the case of the CMOS camera, the average detection rate is 96.3 % with the standard deviation 3.2 %. This difference occurs because of the interlacing effect caused by the frame grabber. The interlacing effect adds harmful variability in the eye appearance.

The detailed experimetnal validation of facial features detection algorithm based on openly accesble face databases is presented in [42].

7.2. Computational time

All procedures, besides binary implementation of SURF were implemented in Matlab. It took 112 seconds to process 439 images from Caltech database which is in average takes 0.25 seconds to process one image. Table 3 presents the computational time taken by each of above discussed steps. For the better representation the time consumed by each step is normalized by the total time necessary for the whole iris detection procedure.

As it can be seen, the most time demanding step is associated with calculating Euclidian distances between descriptors in the knowledge base and descriptors extracted from an input image. This part of the algorithm can be optimized by either using k-d trees or by transforming the training set into a more compact form such as weights of a neural network. It is expected that C implementation of the algorithm will reduce computational time to real time performance.

Table 3. Time taken by each procedure.

PROCEDURE		TIME
Skin-color segmentation		0.042
Facial features segmentation	Feature extraction	0.06
	Feature distances calculation	0.59
	PDF estimation and segmentation	0.07
Detection of Iris & Lips positions		0.136
Miscellaneous		0.102

8. CONCLUSIONS

In this paper an eye movement monitoring system has been presented. The proposed system detects and tracks eyes as well as determines whether the eye is closed or not. For the purpose of eye detection we suggested a skin-color segmentation algorithm based on a neural network and facial feature segmentation algorithm. The facial segmentation algorithm constructs a facial feature model that consists of SURF features. These features are used to estimate class probability density functions. Based on estimated PDFs, pixels are classified into close and open eyes, eyebrows, lips, nose and the rest of face regions. Due to features' scale and brightness invariance and generalization abilities of the algorithm that follows from the dense class features scattering, it robustly segments eye regions regardless of changing eye appearance for different people.

At the last step, localized eyes are tracked with the Extended Kalman filter to prevent interchanging of left and right eye markers. By tracking eyes and knowing their current and previous states we were able to estimate eye closure duration and blinking frequency. Based on these two factors, we can determine whether the driver is drowsy or not.

We implemented the system in Matlab with near real-time performance. We applied the system to the CCD and CMOS videos and the average success rate is 95.3 % with the standard deviation 3.4 % for the CCD and 96.3 % with 3.2 for the CMOS videos.

There are a few directions for further study. One way to reduce computation time is to process the spatial features in a more sophisticated way. Decreasing the number of training vectors by extracting the principle data is one direction to pursue. From the time series of eye positions, we can derive additional information about driver's head movement. For instance we can estimate the head orientation using information about the distance between eyes and the angle of the line interconnecting iris centers.

REFERENCES

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

[2] *The Role of Driver Fatigue in Commercial Road*

Transport Crashes, European Transport Safety Council, 2001.

[3] *Driver Fatigue and Road Accidents*, The Royal Society for the Prevention of Accidents, Birmingham, England, 2001.

[4] L. Hartley, T. Horberry, and N. Mabbott, *Review of Fatigue Detection and Prediction Technologies*, Institute for Research in Safety and Transport, Murdoch University, Western Australia and Gerald Krueger - Krueger Ergonomics Consultants, Virginia, USA, 2000.

[5] S. Box, "New Data from VTTI provides insight into cell phone use and driving distraction," Virginia Tech Transportation Institute, 2009.

[6] S. Zhao and R.-R. Grigat, "Robust eye detection under active infrared illumination," *Proc. of the 18th International Conference on Pattern Recognition*, pp. 481-484, 2006.

[7] Q. Ji and X. Yang, "Real-time eye, gaze, and face pose tracking for monitoring driver vigilance," *Real-Time Imaging*, vol. 8, pp. 357-377, 2002.

[8] J. P. Batista, "A real-time driver visual attention monitoring system," *Proc. of Iberian Conference on Pattern Recognition and Image Analysis*, vol. 3522, pp. 200-208, 2005.

[9] R. I. Hammoud, G. Witt, R. Dufour, A. Wilhelm, and T. Newman, "On driver eye closure recognition for commercial vehicles," *Proc. of SAE Commercial Vehicles Engineering Congress and Exhibition*, Chicago, IL, USA, 2008.

[10] D. Pitts, A. Cullen, and P. Dayhew-Barker, *Determination of ocular threshold levels for infrared radiation cataractogenesis:* NIOSH research report, DHHS publication ; no. (NIOSH) 80-121, DHHS publication - no. (NIOSH) 80-121, 1980.

[11] W. Rong-ben, G. Ke-you, S. Shu-ming, and C. Jiang-wei, "A monitoring method of driver fatigue behavior based on machine vision," *Proc. of Intelligent Vehicles Symposium*, pp. 110-113, 2003.

[12] C. Phil and G. Christos, "A fast skin region detector," *ESC Division Research*, 2005.

[13] U. Tariq, H. Jamal, M. Z. J. Shahid, and M. U. Malik, "Face detection in color images, a robust and fast statistical approach," *Proc. of INMIC*, pp. 73-78, 2004.

[14] A. Hamdy, M. Elmahdy, and M. Elsabrouty, "Face detection using PCA and skin-tone extraction for drowsy driver application," *Proc. of 5th International Conference on Information & Communications Technology*, pp. 135-137, 2007.

[15] D. Butler, S. Sridharan, and V. Chandran, "Chromatic colour spaces for skin detection using GMMs," *Inter. Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 3620-3623, 2002.

[16] I. Naseem and M. Deriche, "Robust human face detection in complex color images," *Proc. of IEEE International Conference on Image Processing*, vol. 2, pp. 338-341, 2005.

[17] O. J. Hernandez and M. S. Kleiman, "Face recognition using multispectral random field texture models, color content, and biometric features," *Proc. of Applied Imagery and Pattern Recognition Workshop*, p. 209, 2005.

[18] C. Chen and S.-P. Chiang, "Detection of human faces in colour images," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 144, pp. 384-388, 1997.

[19] M.-J. Seow, D. Valaparla, and V. K. Asari, "Neural network based skin color model for face detection," *Proc. of Applied Imagery Pattern Recognition Workshop*, pp. 141-145, 2003.

[20] H. Sahbi and N. Boujemaa, "From coarse to fine skin and face detection," *Proc. of the 8th ACM International Conference on Multimedia*, 2000.

[21] A. Lenskiy and J.-S. Lee, "Face and iris detection algorithm based on SURF and circular Hough transform," *Signal Processing, The Institute of Electronics Engineers of Korea*, vol. 47, 2010.

[22] H. Jee, K. Lee, and S. Pan, "Eye and face detection using SVM," *Proc. of Conference on Intelligent Sensors, Sensor Networks and Information*, pp. 577-580, 2004.

[23] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 23-38, 2004.

[24] R. Motwani, M. Motwani, and F. Harris, "Eye detection using wavelets and ANN," *Proc. of GSPx*, 2004.

[25] K. He, J. Zhou, Y. Song, and Q. Qiao, "Multiresolution eye location from image," *Proc. of Signal Processing*, vol. 2, pp. 901-905, 2004.

[26] K.-H. Cheung, J. You, W.-K. Kong, and D. D. Zhang, "A study of aggregated 2D Gabor features on appearance-based face recognition," *Proc. of Int. Conf. on Image and Graphics*, pp. 310-313, 2004.

[27] N. Gourier, D. Hall, and J. L. Crowley, "Facial features detection robust to pose, illumination and identity," *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pp. 617-622, 2004.

[28] Z.-H. Zhou and X. Geng, "Projection functions for eye detection," *Pattern Recognition*, vol. 37, pp. 1049-1056, 2004.

[29] L. Daw-Tung and Y. Chen-Ming, "Real-time eye detection using face circle fitting and dark-pixel filtering," *Proc. of IEEE International Conference on Multimedia and Expo*, vol. 2, pp. 1167-1170, 2004.

[30] H.-J. Kim and W.-Y. Kim, "Eye detection in facial images using zernike moments with SVM," *ETRI Journal*, vol. 30, pp. 335-337, 2008.

[31] A. A. Lenskiy and J.-S. Lee, "Terrain images segmentation in infra-red spectrum for autonomous robot navigation," *Proc. of IFOST 2010*, Ulsan, Korea, pp. 33-37, 2010.

[32] A. A. Lenskiy and J.-S. Lee, "Rugged terrain segmentation based on salient features," *Proc. of International Conference on Control, Automation*

and Systems, Gyeonggi-do, Korea, 2010.

[33] T. D'Orazio, M. Leo, C. Guaragnella, and A. Distante, "A visual approach for driver inattention detection," *Pattern Recognition*, vol. 40, pp. 2341-2355, 2007.

[34] A. A. Lenskiy and J.-S. Lee, "Machine learning algorithms for visual navigation of unmanned ground vehicles," in *Computational Modeling and Simulation of Intellect: Current State and Future Perspectives*, B. Igelnik, Ed., ed: IGI Global, 2011.

[35] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, pp. 346-359, 2008.

[36] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of Association for Computing Machinery*, vol. 15, pp. 11-15, 1972.

[37] K. Deb, A. Vavilin, J.-W. Kim, and K.-H. Jo, "Vehicle license plate tilt correction based on the straight line fitting method and minimizing variance of coordinates of projection points," *International Journal of Control, Automation, and Systems*, vol. 8, pp. 975-984, 2010.

[38] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises and Solutions*, John Wiley & Sons, 1997.

[39] P. Caffier, U. Erdmann, and P. Ullsperger, "Experimental evaluation of eye-blink parameters as a drowsiness measure," *Eur. J. Appl. Physiol*, vol. 89, pp. 319-325, 2003.

[40] G. R. David Dinges, *Perclos: A Valid Psychophysiological Measure of Alertness As Assessed by Psychomotor Vigilance*, Federal Highway Administration, Office of Motor Carriers, Indianapolis, IN, Tech. Rep. MCRT-98-006, 1998.

[41] M. J. Flores, J. M. Armingol, and A. de la Escalera, "Real-time drowsiness detection system for and intelligent vehicle," *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 637-642, 2008.

[42] A. A. Lenskiy and J.-S. Lee, "Detecting eyes and lips using neural networks and SURF features," in *Cross-disciplinary Applications of Artificial Intelligence and Pattern Recognition*, *Advancing Technologies*, Vijay Kumar Mago and N. Bhatia, Eds., ed: IGI Global, 2012.

Artem A. Lenskiy received his Master's degree in Digital Signal Processing and Data Analysis in 2004 from the Novosibirsk State Technical University, Russia and a Ph.D. degree in EE from the University of Ulsan, Korea in 2010. He is currently lecturing at Korea University of Technology and Education (Koreatech), Korea. His research interests include computer vision problems and various applications of processes with long range dependence.

Jong-Soo Lee received his Bachelors degree in Electrical Engineering in 1973 from Seoul National University and his M.S. degree in 1981. In 1985 he was awarded his Ph.D. from Virginia Polytechnic Institute and State University, Blacksburg, USA. He is currently working in the area of multimedia at the University of Ulsan in Korea. His research interests include development of personal English cultural experience programs using multimedia and usability interface techniques to facilitate the acquisition of English language skills by Koreans. He is also working on vocal tract modeling from speech data based on fluid dynamics.