2013 IEEE Intelligent Vehicles Symposium (IV)
June 23-26, 2013, Gold Coast, Australia

# Eye Recognition in Near-Infrared Images for Driver's Drowsiness Monitoring

Bogusław Cyganek, Sławomir Gruszczyński, *Senior Member, IEEE*

*Abstract*— **The paper presents a system for driver's eye recognition from near infrared (NIR) images. The system is organized in a cascade of two classification modules. The first one is responsible for initial eye detection and the second one for eye validation. Detection is based on a novel eye model suited for the NIR images. This process is augmented by the background subtraction module. The subsequent stage of eye validation is performed by the second classifier based on the higher-order decomposition of a tensor with geometrically deformed prototypes. Obtained results in day and night conditions show high accuracy and real-time processing of the system in software implementation.**

## I. Introduction

Drivers' drowsiness and inattention can lead to serious traffic accidents. In this respect, modern computer vision based driver's assisting systems (DAS) can be of help, monitoring driver's state and react to his/her state. However, face and eye observations in car conditions are difficult due to many factors, such as changing illumination, vibrations, dirt, etc.

In this paper an eye recognition system operating entirely in the near infra-red (NIR) conditions is described. It is a modification of our previous system described in [1]. The main modifications are a new background subtraction module, as well as an adaptive eye recognition block.

We also compare the proposed system with some of the systems reported in literature. The first one is a system proposed by Bergasa *et al.* [2]. In this case eye detection relies on a known effect of white spots in NIR images caused by the light reflected by the retina. However, for this purpose the system is equipped with two rings of LEDs. This requires central placement of the camera and the rings. In our system we do not rely on this effect and use only one LED. Another eye recognition system with Kalman and mean-shift trackers was proposed by Zhu and Ji [3]. Their method joins appearance-based object recognition and tracking with active IR illumination, using the mentioned effect of high reflectance of the pupil. Eye classification is done with the SVM. The method allows real-time eye detection and tracking under variable lighting conditions and various orientations of faces. However, the system was not designed to operate in a moving car.

A three stage drowsiness detector is proposed by García *et al.* [4]. At first face and eye are detected with the method by Viola and Jones [5]. Then the Kalman filter is employed for tracking. Positions of the pupils are then detected and combined with the adaptive lighting filtering. Finally, the percentage of eye closure parameter (PERCLOSE) is computed which can indicate driver's state.

A system for driver inattention detection based on eye recognition from visible spectrum images was proposed by de Orazio *et al.* [6]. First, iris regions are detected with the Hough transform. Then eye candidates are extracted analyzing symmetrical regions. These are next validated with the back-propagation neural network trained with the wavelet signals of the off line collected eye and non-eye prototype patterns. Driver's conditions are then inferred by the probabilistic model, built upon the multivariate Gaussian mixture that characterizes normal behavior of a driver. For this purpose the percentage of time the eyes are closed and eye closure frequency are used. Parameters of the Gaussian mixture are computed with the expectation-minimization method. As reported, their system attained 95% detection rate and operates in near real-time on 320×240 video from the webcamera mounted in a car. However, it is limited only to the daylight conditions.

The paper is organized as follows. Architecture of the proposed system is presented in the next section. Then the building blocks are described, starting from the background detection module, up to the final classifier. The paper ends with experimental results and conclusions.

## II. Structure of The System

Figure 1 depicts our system for driver's eye recognition assembled in the test cars. The system consists of two cameras for the visible and the NIR spectra, as well as one LED for NIR illumination. The system can operate in day and night conditions.

Figure 1. Presented system for driver's eye recognition mounted in the cars. The acquisition system contains two cameras, one for the visible and one for the NIR spectrum. NIR lighting is obtained with the external LED.

Figure 2 presents block diagram of the system. As shown, processing is done in two stages. First eye region candidates are detected. Then these are sifted out by the eye verification module based on the higher-order decomposition of tensors containing prototype eye and non-eye patterns. Detection is

Authors are with the AGH University of Science and Technology, Al. Mickiewicza 30, Krakow 30-059, Poland (corresponding author e-mail: cyganek@agh.edu.pl).

augmented with the background model computed from the consecutive frames.

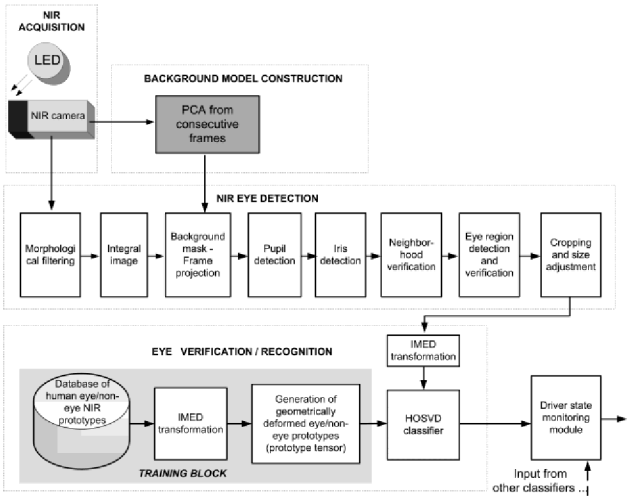


Figure 2. Block diagram of the eye recognition system. Processing is carried out in two stages. First eye region candidates are detected. Then these are sifted out by the eye verification module based on higher-order decomposition of the prototype tensors. Detection is constrained to the moving areas found with the background model computed from the consecutive frames.

Processing starts with morphological erosion of an input image. Then the integral image is created which simplifies computation of sums of pixel values in any rectangular area of the input image. In parallel a background model is constructed which is used to create a mask of moving regions. These greatly facilitate detection since eyes, lips, and other regions of a driver's face are moving. This process is described in the next section. After that pupil and iris regions are detected based on the proposed eye model. Finally, neighboring regions are checked to validate a candidate region. These are then fed to the classifier trained with deformed prototypes of eye and non-eye patterns. From the patterns a tensor is formed which is then decomposed with the higher-order singular value decomposition (HOSVD) to build a subspace representing the training patterns. These issues are presented with details in our previous paper [1]. In this paper we focus on new additions to these, i.e. the background subtraction and detection modules while briefly presenting the other for better comprehension.

## III. System Operation

In this section the building blocks of the system are described with stress on new modules. Details on the other ones can be found in [1].

### A. Adaptive Background Subtraction

Background subtraction helps in detecting moving parts in a video sequence. The simplest approach consists in subtracting two frames and thresholding on values different from 0. However, this simple method does not work well in practice because of local pixel variations and noise. Therefore many other methods were proposed, such as the ones relying on a statistical model of the static background with the mixture of Gaussians or the mean-shift [7].

One of the methods that offer good results with reasonable computation time was proposed by Oliver *et al.* and is called eigenbackground [8]. The method relies on the PCA model of a static background computed from a chosen sequence of frames. The idea is to describe static regions of the scene with the eigenspace. On the other hand, new images with regions that differ from the model are not perfectly reconstructed by this model. This information can be used to discover moving objects in a frame $I_k$. For this purpose it is first projected into the already created eigenspace. Then the reconstructed frame $I_{k+1}$ is obtained. The last step is to compute and threshold the Euclidean distance between the two frames. We briefly outline details of this method since it is used in our system.

Let us recall that for data points represented with $N$ vectors $\mathbf{x}$, each of dimension $D$, PCA consists in finding parameters $\mathbf{T}$ and $\mathbf{\Lambda}$ of the following decomposition

$$\mathbf{A}\underbrace{E\left(\left(\mathbf{x}-\mathbf{m}_x\right)\left(\mathbf{x}-\mathbf{m}_x\right)^T\right)}_{\mathbf{\Sigma}_x}\mathbf{A}^T=\Lambda\,, \tag{1}$$

where $E$ denotes expected value and $\mathbf{m}_x$ is an average of all $\mathbf{x}$.

However, for a large dimensional data space a direct computation of (1) can be time and memory consuming, since the covariance matrix is of dimensions $D\times D$. In our case $D$ is high since this is a total number of all pixels in a single frame. For instance, for a frame of resolution 640×480, the corresponding covariance matrix would be of size 307200×307200 which is rather prohibitive. Even more problematic is the eigendecomposition of such a matrix. However, in the case of much lower number of images $N$ (in our case these are 3-7), computations can be greatly reduced if constrained to the problem of finding only $N$ eigenvectors of a much smaller matrix [9]. For this purpose from the zero-mean points $\bar{\mathbf{x}}_i$ a $D\times N$ column matrix $\bar{\mathbf{X}}$ is created. Then, its low dimensional $N\times N$ product matrix $\bar{\mathbf{X}}^T\bar{\mathbf{X}}$ is computed, which next can be eigendecomposed in accordance with (1). In effect, the following is obtained

$$\bar{\mathbf{X}}^T\bar{\mathbf{X}}\,\mathbf{e}_k=\lambda_k\,\mathbf{e}_k\,, \tag{2}$$

where $\mathbf{e}_k$ are respectively eigenvectors and $\lambda_k$ eigenvalues of the matrix $\bar{\mathbf{X}}^T\bar{\mathbf{X}}$. Now, left multiplying by $\bar{\mathbf{X}}$ both sides of the above the following is obtained

$$\underbrace{\bar{\mathbf{X}}\bar{\mathbf{X}}^T}_{\mathbf{\Sigma}_x}\underbrace{\left(\bar{\mathbf{X}}\,\mathbf{e}_k\right)}_{\mathbf{q}_k}=\lambda_k\underbrace{\left(\bar{\mathbf{X}}\,\mathbf{e}_k\right)}_{\mathbf{q}_k}. \tag{3}$$

From (3) it is easy to notice that

$$\mathbf{w}_k=\bar{\mathbf{X}}\,\mathbf{e}_k\,, \tag{4}$$

are eigenvectors of dimensions $D\times1$ of the covariance matrix which is a product of two matrices, as follows

$$\mathbf{\Sigma}_x = \bar{\mathbf{X}}\bar{\mathbf{X}}^T . \quad (5)$$

Thus, if $\mathbf{e}_k$ are known, then based on (4) the matrix of eigenvectors of $\mathbf{\Sigma}_x$ can be computed simply as follows

$$\mathbf{P} = \bar{\mathbf{X}}\,\mathbf{E}, \quad (6)$$

where $\mathbf{E}$ is a matrix with columns of $\mathbf{e}_k$. Finally, let us notice that eigenvectors in $\mathbf{P}$ need to be normalized to form an orthonormal basis. Thus, $\mathbf{A}$ in (1) is obtained as follows

$$\mathbf{A} = \bar{\mathbf{P}}^T , \quad (7)$$

where $\bar{\mathbf{P}}$ is a column-wise normalized matrix $\mathbf{P}$. Based on this a much faster procedure for computation of the eigendecomposition is obtained. The background subtraction algorithm is summarized in Figure 3.

**1.** From the consecutive frames compute PCA decomposition;

**2. C**ompute a projection $I_{k+1}$ of a frame $I_k$ onto the eigenspace found in step 1;

**3.** Reconstruct the projected image $I_{k+1}$ to obtain $I_{k+2}$;

**4**. Create the background map $B$ as follows:

$$B_i = \begin{cases} 0, & for\,|I_{k+1} - I_{k+2}| < \rho \\ 1, & otherwise \end{cases} \quad (8)$$

where $i$ denotes pixel index and $\rho$ is a threshold;

Figure 3. Eigenbackground subtraction algorithm.

Results of this procedure on exemplary frames from the NIR signal are presented in Figure 4 and Figure 5.

In the proposed system the computed eigenbackground images are used to guide the eye detection process since eye regions are assumed to belong to the moving parts of the scene. We observed that the parameter $\rho$ can be set depending on an average intensity of the input image. In our experiments its value was set in the range 20-40, the smaller value for bright and the upper for dark images, respectively.

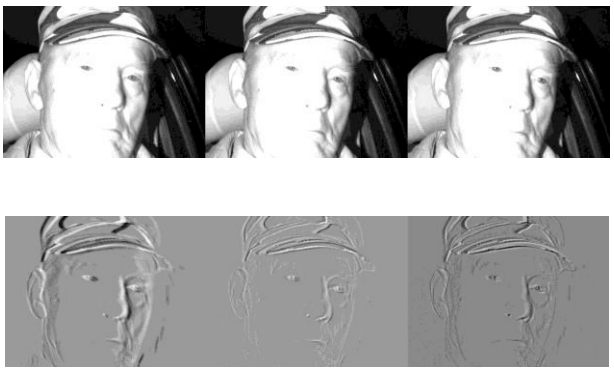

Figure 4. A PCA background model built from the three consecutive frames of a driver (upper row). Computed eigenimages (lower row).

Figure 5. Eigenbackground subtraction. Test frames in upper row. Eigenbackground in lower row. White areas correspond to moving parts of the scenes.

Thanks to the aforementioned procedure, computation of the eigenbackground model is relatively fast and in our setup requires 0.025 s on average for the 640x480 images. This process is done in a separate thread, as shown in Figure 2.

### *B. Initial Eye Detection*

For initial eye detections in the NIR images a simple eye model is proposed, as depicted in Figure 6 [1].

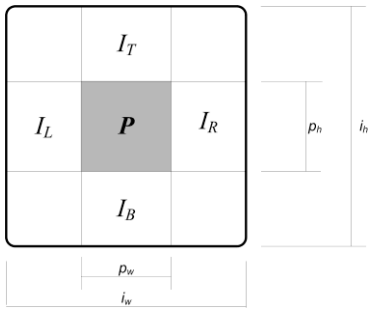


Figure 6. Eye model for detection in NIR images. ***P*** denotes pupil area. The regions $I_L$, $I_T$, $I_R$, $I_B$ correspond to different parts of the iris.

In this model a pupil is represented as a square central region $\boldsymbol{P}$ of dimensions $p_w \times p_h$ (Figure 6).

**1.** Erode a frame with the $m_w \times m_h$ structural element
**2.** Compute the integral image;
**3.** For each pixel do:
**4.** If an average intensity in a pupil area $P$ is below the threshold $t_p$, then:

$$s_P = S_{av}\left(P\right) < t_p \tag{9}$$

**5.** If a ratio of an average intensity of iris to an average intensity of the pupil is within the range $R_{ip}$, then:

**6.** If the mutual ratios of average intensities in the iris region $I_T, I_B, I_L, I_R$

$$\frac{\left|S_{av}\left(I_a\right) - S_{av}\left(I_b\right)\right|}{S_{av}\left(I_a\right) + S_{av}\left(I_b\right)} < r_{ab} \tag{10}$$

are below the thresholds $r_{LR}$, $r_{BT}$, $r_{TL}$, $r_{BR}$ for all $a,b \in \{T,B,L,R\}$, then:

**7.** Insert the region to the queue ordered by the values of $s_P$ in (9);

**8.** Remove overlapping eye regions from the output;

Figure 7. Initial eye detection algorithm.

Values of $p_w$ and $p_h$ were obtained experimentally. The most important parameters are shown in TABLE I. On the other hand, iris is modeled by the four separate rectangular regions $I_T$, $I_B$, $I_L$, $I_R$, for top, bottom, left, and right iris regions, respectively (Figure 6). Their dimensions are set in accordance with dimensions of the pupil and the $i_w$ and $i_h$ which define width and height of the iris region, respectively (TABLE I). More details are presented in Figure 8 and are discussed in [1].

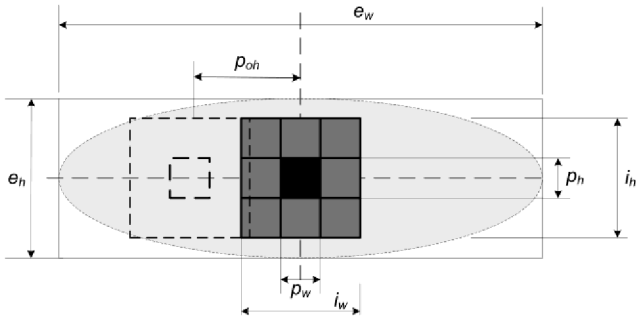


Figure 8. Parameters of an eye model.

The pupil candidates are found by computing an average intensity of each region of size $p_w \times p_h$ in the filtered NIR image (Figure 2). These are then checked if these average values are less than a chosen threshold $t_p$. In the next step, a ratio of an average intensity of an iris to an average intensity of the pupil areas is checked. If that ratio is in the predefined range then the regions are further processed to verify mutual relations between pairs of the iris areas. All computations of sums of pixels in rectangular areas are fast due to the precomputed integral image. Thanks to this, for any dimensional rectangle a sum of all its pixels can be computed with only four additions. Details of computations are discussed in [12]. At the end, if all of the above conditions are fulfilled, then such a region is entered to the priority queue which is ordered by the value of the average pupil intensity $s_P$ computed in (9). That is, the lower the pupil intensity the stronger position in the queue. Steps of the initial eye detection are summarized in the algorithm in Figure 7.

TABLE I. THE MOST IMPORTANT PARAMETERS OF THE NIR EYE DETECTOR (SEE FIGURE 8)

| Parameter | Meaning | Values |
|---|---|---|
| $p_w$ | Pupil width | 11-13 pixels |
| $p_h$ | Pupil height | 11-13 pixels |
| $i_w$ | Iris width | [0.85-1.1] * $p_w$ |
| $i_h$ | Iris height | [0.85-1.1] * $p_h$ |
| $e_w$ | Eye width | [6.9-9.7] * $p_w$ |
| $e_h$ | Eye height | [3.1-4.7] * $p_h$ |
| $p_{oh}$ | Horizontal pupil distance | 2 * $p_w$ |

### C. Eye Validation

The next module in the classification chain shown in Figure 2 consists of the classifier operating with geometrically deformed eye and non-eye prototypes. The main idea is that for each prototype pattern, a tensor is constructed which contains artificially generated expected views of that pattern. Each such tensor is then decomposed with the HOSVD method to span a space of orthogonal base tensors, as proposed by Savas and Eldén [10]. Recognition consists in checking a distance of the projections of the test pattern in the built tensor spaces. The smallest distance indicates a class of an object. However, before processed, the patterns are additionally transformed to take advantage of the Euclidean distance metrics proposed by Wang *et al.* [11].

In the following we take an approach that tensors are multidimensional arrays of real values. As already mentioned, important information on tensor contents can be obtained by its HOSVD decomposition [13][14]. Thus, a deformed prototypes tensor $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_P}$ with dimensions $N_1, N_2, \ldots, N_P$ can be represented as follows

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_P \mathbf{S}_P, \tag{11}$$

where $\times_j$ denotes the $j$-mode tensor matrix product, $\mathbf{S}_k$ are unitary mode matrices of dimensions $N_k \times N_k$, and $\mathcal{Z} \in \Re^{N_1 \times N_2 \times \ldots N_m \times \ldots N_n \times \ldots N_P}$ is a core tensor [13][14].

For a given mode matrix $\mathbf{S}_i$ in (5), the HOSVD decomposition leads to the set of orthogonal base tensor subspace which is used for eye recognition. A distance of a projection of a test patterns $\mathbf{P}_x$ to the subspace is given as follows [10]

$$\hat{\rho}_k = \sum_{h=1}^{N_k} \left\langle \hat{\mathcal{T}}_h^k, \hat{\mathbf{P}}_x \right\rangle^2, \tag{12}$$

where

$$\mathcal{T}_h = \mathcal{Z} \left( \underbrace{:,:,\ldots,:}_{P-1}, h \right) \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_{P-1} \mathbf{S}_{P-1} \tag{13}$$

are the base tensors and $\mathbf{s}_P^{(h)}$ denotes an $h$-th column of the unitary matrix $\mathbf{S}_P$. The index $h$ in (12) spans the possible values of the $P$-th dimension, i.e. $1 \le h \le N_P$.

It is assumed that each prototype tensor is used to build a separate space. Thus, for $K$ available training patterns a number of deformable tensors $\mathcal{T}^k$ are created, in which $1 \le k \le K$ denotes a class (i.e. a different training object). Then, a test pattern $\mathbf{P}_x$ is checked with all $\hat{\mathcal{T}}_h^k$ in accordance with (12). A value of $N_k$ denotes a number of chosen components, which is a parameter of this classification method.

In the case of a positive recognition of exactly two eyes there is a final stage which checks their geometrical relation. For instance it is verified whether two regions do not overlap or are not too distant apart [1].

## IV. Experimental Results

The system was implemented in C++ using the HIL library [12] which helps in the low level image operations. Experiments were run on the computer equipped with the 8 GB RAM, eight-core Pentium® i7 Q 820 processor @ 1.73GHz, and 64-bit Windows® 7 OS. Additionally, time intensive procedures were accelerated with the OpenMP library to take advantage of the multi-core processors. Thanks to this a speedup ratio of up to three times were obtained compared to the pure serial implementation. A number of 10-20 minutes long video sequences were recorded in which participated volunteer drivers. The whole system mounted in a car is shown in Figure 1. It can become a part of an intelligent DAS [15].

Figure 9 depicts initial eye detection in an exemplary test sequence. Left image contains all pupil candidates. Candidate eye regions which are fed to the HOSVD eye recognition module for final validation are shown in the right image in Figure 9.

Figure 9. Example of eye detection in a test sequence. Left image contains all pupil candidates. Candidate eye regions which are fed to the HOSVD eye recognition module for final validation are shown in the right image.

As already mentioned, initial detection is greatly improved thanks to the mask obtained from the eigenbackground model. Since the mask corresponds to the areas which changed position, only these are checked for eye candidates. An example of this process is presented in Figure 10.

Figure 10. Moving parts of a driver detected with the eigenbackground subtraction method.

Figure 11 shows results of the next step of eye recognition in one of the test sequences. Eye regions that passed verification by the HOSVD classifier, i.e. the ones that have been recognized as eyes, are shown in black frames.

As in all classification systems we aim at highest recognition accuracy. However, in the case of driver's drowsiness monitoring based on eye analysis, a very important factor is a percentage of false-positive answers since these if present can give false information that a driver is awaken whereas it is not a case. Therefore parameters of the whole classification chain were set to obtain the FP ratio as small as possible. Average accuracy parameters of the eye recognition in the ten test sequences are summarized in TABLE II.

TABLE II. Average Accuracy Of Eye Detection

| TP | FP | FN |
|---|---|---|
| 97.4 | 0.1 | 2.5 |

When compared with our previous system described in [1], in this version thanks to the described improvements it was possible to increase TP by 0.5-1.5% and to lower FP by a factor or 0.02-0.05%.

Figure 11. Results of final eye recognition in a test sequence. Eye regions that passed the HOSVD classifier are framed.

A number of eyes detected in each frame is then used to create a PERCLOSE diagram [1]. Periods of open eyes are encoded with value 2. Partially open eyes, or only one eye visible are marked with 1. Finally, closed eyes periods are at encoded by 0. These can be used to infer on driver's drowsiness, as described in literature [4][1].

## V. Conclusion

The paper presents a NIR based system for driver's eye recognition, capable of operating in real car conditions and different illuminations. Thanks to the only one NIR illuminating LED mounted on the rail attached to the windshield, the system is safe for a driver. Compared to the previous version of the system presented in [1], in this paper we focused on its modifications which increased accuracy. The eigenbackground subtraction module was built into the system which allows detection of moving parts of a driver. Thanks to this the initial eye search process is greatly reduced only to the moving areas. The eigenmodel of the background is built in parallel to the frame processing. For further research a dynamic retraining of the HOSVD classifier is considered. New eye patterns coming from the positively recognized regions in the previous frames could be used to retrain this classifier. Thanks to this, the classifier can adapt to the local conditions. Also, further tests are conducted with more volunteer drivers and more cars, as well as in different conditions. One of the goals is to develop algorithms of automatic setup of control parameters.

## Acknowledgment

The authors would like to express their gratitude to Mr. Marcin Bugaj for his help in the experiments. The work was partially supported by the National Center for Research and Development under Lider Program, contract no. LIDER/16/66/L-1/09/NCBiR/2010, as well as in part by the Polish National Science Centre NCN contract no. DEC-2011/01/B/ST6/01994.

## References

[1] B. Cyganek, S. Gruszczyński, "Hybrid Computer Vision System for Drivers' Eye Recognition and Fatigue Monitoring", Neurocomputing, (accepted for publication), 2012.

[2] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, E. Lopez, "Visual Monitoring of Driver Inattention, In D. Prokhorov (Ed.): Computational Intelligence in Automotive Applications, SCI 132, 2008, pp. 25–51.

[3] Z. Zhu, Q. Ji "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," Computer Vision and Image Understanding 98, 2005, pp. 124–154.

[4] I. García, S. Bronte, L. M. Bergasa, J. Almazán, J. Yebes, "Vision-based Drowsiness Detector for Real Driving Conditions," 2012 Intelligent Vehicles Symposium. Alcalá de Henares, Spain, 2012.

[5] P. Viola, M. Jones, "Robust real-time face detection," Proceedings of the International Conference on Computer Vision, pp. II. 747-755, 2001.

[6] T. de Orazio, M. Leo, C. Guaragnella, A. Distante, "A visual approach for driver inattention detection," Pattern Recognition 40, 2007, pp. 2341–2355.

[7] M. Piccardi, "Background subtraction techniques: a review," 2004 IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, 2004, pp. 3099-3104.

[8] N. M.Oliver, B.Rosario, A. P.Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions, " IEEE PAMI, Vol. 22, No. 8, 2000, pp. 831-843.

[9] M.A. Turk, A.P.Pentland, "Face recognition using eigenfaces," IEEE Conf. on Computer Vision and Pattern Recognition, 1991, pp. 586-590.

[10] B. Savas, L. Eldén, "Handwritten digit classification using higher order singular value decomposition", Pattern Recognition, Vol. 40, 2007, pp. 993–1003.

[11] L. Wang, Y. Zhang, J. Feng, "On the Euclidean Distances of Images," IEEE PAMI, Vol. 27, No. 8, 2005, pp. 1334-1339.

[12] B.Cyganek, J.P. Siebert, "An Introduction to 3D Computer Vision Techniques and Algorithms", Wiley, 2009.

[13] L. de Lathauwer, "Signal Processing Based on Multilinear Algebra," PhD dissertation, Katholieke Universiteit Leuven 1997.

[14] L. de Lathauwer, Moor de, B., Vandewalle, J. "A Multilinear Singular Value Decomposition," SIAM J. Matrix Analysis and App., Vol. 21, No. 4, 2000, pp. 1253–1278.

[15] Tadeusiewicz R. "Introduction to Intelligent Systems", Chapter in the book: Wilamowski B.M., Irvin J.D. (Eds.): The Industrial Electronics Handbook – Intelligent Systems, CRC Press, Boca Raton, 2011, pp. 1-12.