Hybrid computer vision system for drivers' eye recognition and fatigue monitoring

Article *in* Neurocomputing · February 2014

DOI: 10.1016/j.neucom.2013.01.048

2 authors:

Boguslaw Cyganek
AGH University of Science and Technology in Kraków
185 PUBLICATIONS **1,714** CITATIONS
SEE PROFILE

Sławomir Gruszczyński
AGH University of Science and Technology in Kraków
265 PUBLICATIONS **2,124** CITATIONS
SEE PROFILE

Hybrid Computer Vision System for Drivers' Eye Recognition and Fatigue Monitoring

Bogusław Cyganek, Sławomir Gruszczyński

AGH University of Science and Technology
Al. Mickiewicza 30, 30-059 Kraków, Poland
cyganek@agh.edu.pl

Abstract. The paper presents a hybrid visual system for monitoring driver's states of fatigue, sleepiness and inattention based on driver's eye recognition. Safe operation in car conditions and processing in daily and night conditions are obtained thanks to the custom setup of two cameras operating in the visible and near infra-red spectra, respectively. In each of these spectra image processing is performed by a cascade of two classifiers. The first classifier in a cascade is responsible for detection of eye regions based on the proposed eye models specific to each spectrum. The second classifier in each cascade is responsible for eye verification. It is based on the higher order singular value decomposition of the tensors of geometrically deformed versions of real eye prototypes, specific to the visible and NIR spectra, respectively. Experiments were performed in real car conditions in which participated four volunteer drivers. The obtained results show high recognition accuracy and real-time processing in software implementation. Thanks to these the system can become a part of the advanced driver's assisting system.
Keywords: driver monitoring, eye recognition, tensor processing

1 Introduction

Road accidents are the leading cause of death in industrialized countries. These are mostly caused by human errors occurring in different activities related to vehicle driving, as well as low violations such as speeding. Also drivers' fatigue, sleepiness or inattention can lead to serious accidents. As reported by the European Union only driver tiredness constitutes a significant factor of about 20% of crashes involving heavy commercial vehicles [29]. These problems were recognized and countermeasures were undertaken in many countries. For instance, in the European Union there are plans to significantly reduce a number of fatal accidents in the nearest years by changing low regulations and also employing new technologies [28][30].

Thanks to recent hardware and software developments computers can be used to monitor drivers' conditions with potential of alerting in dangerous situations [45][5]. In this respect hybrid intelligence offers new methods and tools which can be successful in solving such difficult problems, as reported in literature [11][1][20] [60][12][46][8]. Such driver's supporting methods become parts of the Driver Assisting Systems (DAS). It is a matter of incoming years to have DAS practically available in modern cars. One of the key functionalities of DAS is reliable detection of driver's fatigue and sleepiness, mostly based on observation of his/her eyes.

Face and eye recognition belong to the fundamental tasks of computer vision [27][56][57]. Eye recognition for monitoring driver's behavior, fatigue, sleepiness or inattention is one of the research topics which were greatly enhanced for the last years. Eye detection can be divided into active and passive methods, respectively [58]. In the former group, a custom hardware is required which supplies special lighting, such as near IR [61][5]. On the other hand, passive methods assume only the natural illumination spectrum. This group can be further partitioned into the methods which use color or monochrome images, respectively [7]. Reported eye recognition methods rely on detection of some characteristic eye features. These, in turn, can be obtained with template matching [33], projections [62], Hough transform [45][62], gradients [36] or wavelets [44], to name a few. Frequently used indicator of driver's state is the percentage of eye closure (PERCLOSE) [23][24][54]. However, the existing solutions still lack sufficient accuracy or speed to reliably operate in real car conditions.

In this paper a visual system for driver's eye recognition is proposed which allows real-time operation in moving vehicles and under various conditions. For this purpose the original hardware setup was constructed which consists of two cameras operating in the visual and near infra-red (NIR) spectra. NIR illumination is obtained with a NIR light emitting diode (LED). The paper follows our previous work focused upon a method of processing images acquired in only daily conditions with color camera [18]. The system presented in this paper contains a new vision path processing NIR signals which allows operation in night conditions.

In the presented system there are two independent blocks of visual signal processing. Each of them is composed of two different classifiers connected in series: The front-end is responsible for detection of eye candidates, followed by the trained classifier which role is to refine the candidates and to respond whether the eyes are well visible. The two front detectors for the visible and NIR spectra are different, however. In the former, operation starts with the skin segmentation from color images [18]. This stage is followed by the skin

region detection obtained with the adaptively window growing method (AWG) [14]. Then, eye candidate regions are selected thanks to the proposed eye model. Finally, eyes are verified by the classifier operating with the higher order singular value decomposition (HOSVD) of the tensor of geometrically deformed images of real eye prototypes [15]. On the other hand, processing of the NIR images starts with detection of pupil candidates. This is done in the associated integral image which allows very fast computation of sums of pixel values in any rectangle within an image. Then iris and finally eye regions are detected based on the novel eye model, designed especially for the NIR signals. Eye regions are then refined to only those which fulfill predefined geometrical relations. Finally and similarly to the visible spectrum path, eye region candidates are recognized by the HOSVD classifier, this time trained with the prototypes from the NIR images, however. In this paper we focus mostly on the new NIR processing path. As already mentioned, part of the system operating in the visible spectrum is described in our previous paper [18].

Driver's safety is especially stressed in the presented system. This relates to the hardware construction and especially to the NIR lighting module. It is well known that long exposition to the NIR illumination of high power, although not directly visible, can be annoying or even dangerous for the exposed person [2]. Therefore in our system we use only one LED of low power synchronized with the shatter of the camera.

Experiments were conducted in different car conditions with help of four volunteer drivers. Presented experimental results obtained in real car conditions show high accuracy and real-time operation of the system in software implementation.

The paper is organized as follows. Architecture of the system and the custom hardware setup are presented in section (2). The proposed method of eye detection in NIR spectrum is dealt with in section (3). Similarly, detection in the color images is discussed in section (4). In section (5) the method of eye recognition in the tensor space of deformed pattern prototypes is presented and discussed. Additionally, the Image Euclidean Distance method is presented which increases accuracy of eye recognition. Experimental results are presented in section (6). The paper ends with conclusion in section (7) as well as with the literature references.

2 System Architecture

Block diagram of the proposed system is presented in Figure 1. It is organized into two distinctive paths, one for processing images from the visible spectrum, the second from the NIR spectrum, respectively. Further, each processing path is organized as a cascade of specialized modules, each refining output of its predecessor. Such connections of many classifiers are characteristic of usually higher accuracy compared to complex but simple classifiers [37][34][48]. As already mentioned, in this paper we mostly focus upon description of the NIR module.

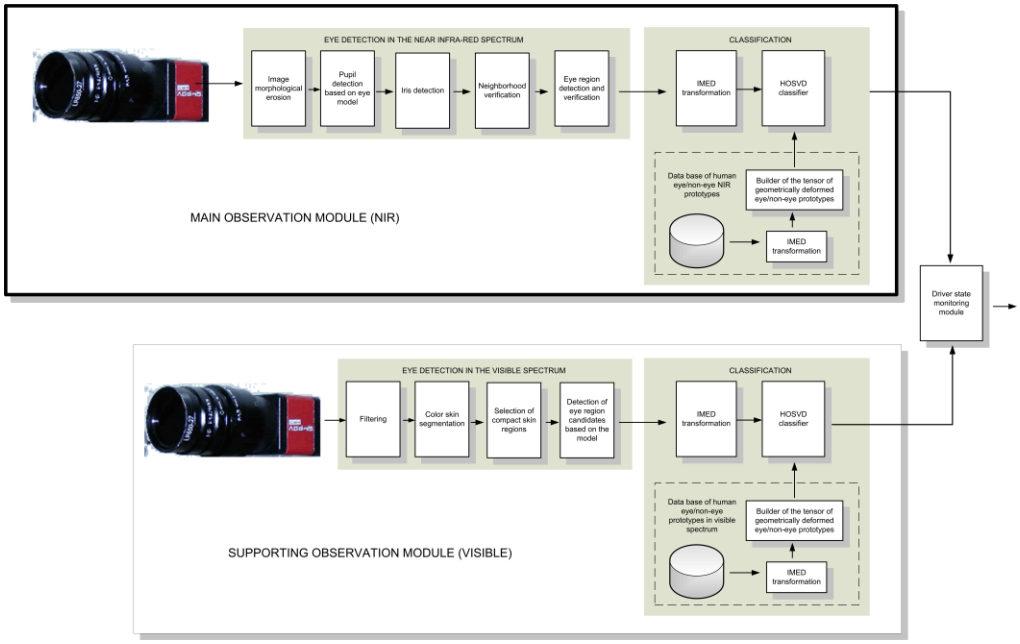


Figure 1. Architecture of the eye recognition system. Two signal processing modules are used which process visible and NIR spectra from the two cameras. Information from both is gathered and processed by the driver state monitoring module.

An overview of the NIR processing path is as follows. Eye detection starts with acquisition of the NIR images by the camera equipped with the optical filter disabling visible light from passing into its sensor, as described in

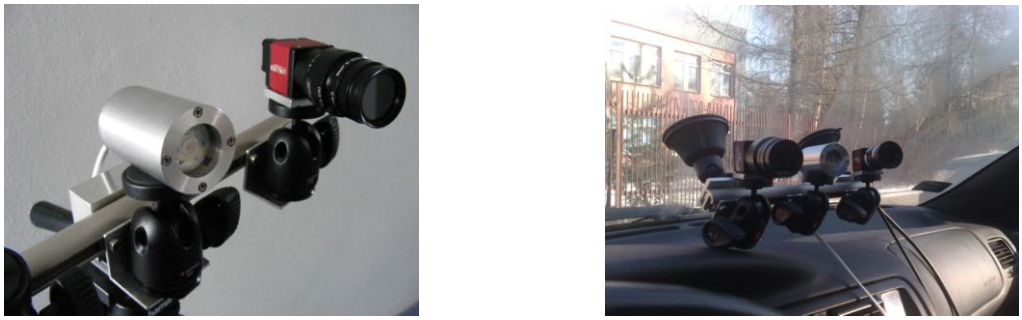
the next section. Then images are nonlinearly processed with the morphological filter. This removes the bright spot from the iris reflecting the NIR illumination, as well as noise in the images. Then eye detection is performed based on the proposed eye model. First detected are pupils, then iris and based on these eye candidate regions are output. These are then filtered out removing overlapping regions, as well as the ones which compliance with the model is not high. Details of this process are presented in section (3). This way detected eye region candidates are passed to the second classifier which role is to recognize open eyes if presented in the image. As already mentioned, this classifier is based on the HOSVD decomposition of prototype tensors, hence it is referred to as the HOSVD classifier. However, the module is preceded by the Euclidean Image Distance transformation which role is to pre-process the patterns into the more discriminative versions which takes into account mutual relations of pixel positions, as discussed in section (5.1). The HOSVD classifier is trained with the eye and non-eye prototypes gathered from real images. Both processing paths employ the same type of the HOSVD classifier trained with different sets of patterns, however.

Processing of the images from the visible spectrum is done with color image acquisition, followed by the low-pass filtering to remove noise. Then the skin regions are segmented based on the set of fuzzy rules defining subset of characteristic colors of human skin. These are then grouped together by the adaptive window method. In the obtained skin regions eyes are detected based on our eye model, which is different from the model used for detections in the NIR images. Then eye region candidates are classified by the already discussed HOSVD classifier. Further details on processing of the color images are presented in our previous publication [18].

The two outlined paths of visual signal processing are fed to the module which infers on driver's state which is not addressed in this paper. However, the presented blocks can operated independently providing information on driver conditions.

2.1 Hardware Specification

In this section we present details of hardware setup used for acquisition of the color and NIR images, as presented in Figure 1. Image acquisition is done by a setup consisting of the two cameras and LED which does NIR illumination, as shown in Figure 2.



a b

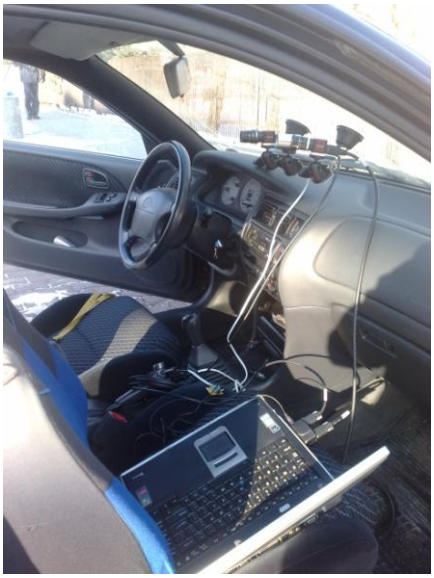
Figure 2. The experimental setup mounted on a tripod (a). Visible is the dovetail rail with cameras and illumination LEDs attached with the arm mounting devices equipped with the ball-pivot connectors. The setup with two cameras and LED illumination mounted to the windshield of a car with two sucker connectors (b).

The used cameras are Allied Vision® Guppy PRO F-031C and F-032B models, operating in the visible and NIR spectra, respectively [4]. The Guppy PRO F-031B/C is a small dimensions 1394b camera with the ICX618 sensor. Its ExView HAD CCD sensor is very sensitive both in the visible and in the NIR spectra. Therefore the NIR camera is equipped with the sharp optical bandpass filter attached to its lens which stops the visual band from passing to the camera sensor.

Figure 2a shows the experimental setup mounted on a tripod. Visible is the dovetail rail with attached cameras and illumination LEDs. The arm mounting devices are equipped with the ball-pivot connectors which allow settings of the cameras and LED in any direction. Figure 2b shows this hardware setup mounted to the windshield of a car with two suction pivot connectors.

As alluded to previously, NIR LED illumination is used in all systems relying on NIR detection. However, very important (and sometimes overlooked) problem is driver's safety when exposed on long NIR lighting [2]. For instance, Bergasa *et al.* address this problem although their system needs to contain two rings of NIR LEDs [5]. As reported, such a setup was necessary to obtain the effect of very light iris spots when lighted with the NIR source. However, based on our experiments a long exposure to the NIR illumination, even if fulfilling the formal regulations regarding the allowed power [2], can be very tiring to a driver. Therefore in the presented

system we use the low power NIR illumination with only one synchronized LED. Its power was also set to a minimal level which allows proper exposition.



a

b

c

Figure 3. The driver's monitoring system mounted in a car (a). View of the camera-illumination setup connected through the IEEE-1394 links to the laptop computer with the driver monitoring software (b). A driver and the monitoring system in a car (c).

Figure 3a depicts our driver's monitoring system mounted in a car. View of the camera-illumination setup connected through the IEEE-1394 links to the laptop computer with the driver monitoring software is shown in Figure 3b. One of the volunteer drivers and the monitoring system in a car are visible in Figure 3c.

2.2 Software Implementation and Computation Platform

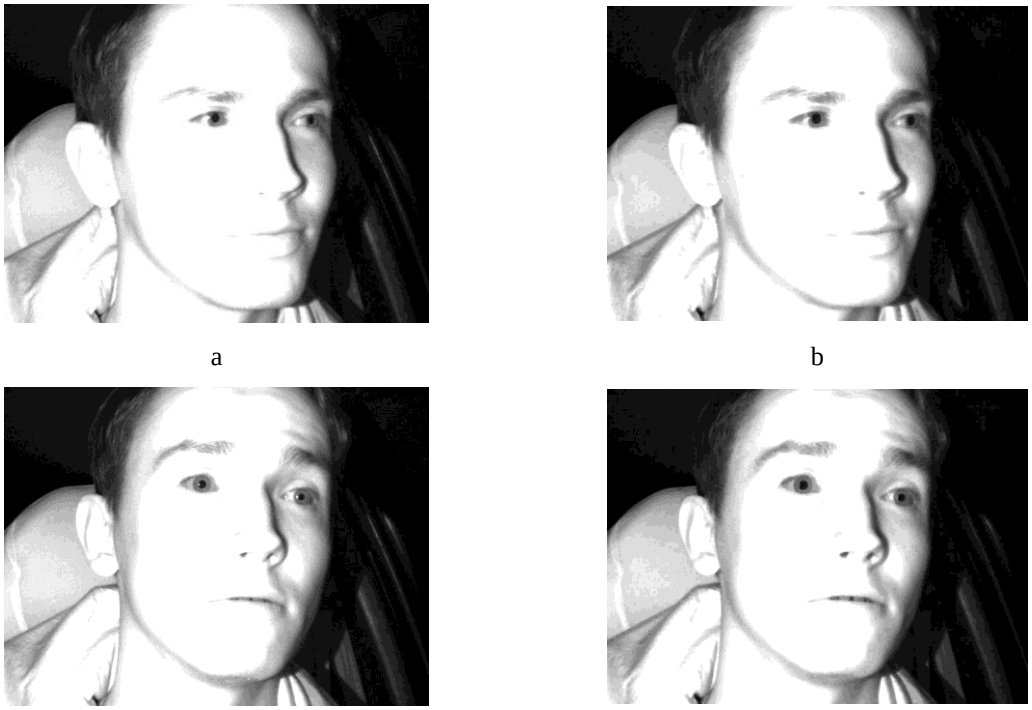
Software of the system was implemented entirely in C++ using the HIL library [31]. The library facilitates low level image operations [14]. Majority of these operations were implemented with the OpenMP framework to take advantage of the multi-core systems, as described in [19]. For tensor processing the C++ software component proposed in [17] was used. Experiments presented in section (6) were run on the computer equipped with 8 GB RAM, eight-core Pentium® i7 Q 820 processor @ 1.73GHz, supervised with the 64-bit Windows® 7 operating system.

3 Eye Detection in the Near Infrared Images

As already mentioned, detection of eye candidates is performed with help of a proposed here eye model. However, it is important to notice that because of the cascade configuration of classifiers its role is to sieve out only the regions which definitely are not eye regions, passing at the same time all those regions which, even with some doubts, can contain the eyes. More specifically, at first we care about small false negatives (FN) ratio and then about the high true-positive (TP). In other words, it is better for the detector to pass a region which by the next module will be rejected as not an eye, rather than to reject a true eye region. The last requirement on the detector is to process the frames in real-time which also influences the chosen eye model and the algorithms.

Eye detection in NIR differs from detection in the visible spectrum at daily conditions. The main difference is presence of an artificial source of the NIR radiation which reduces dependability on external lighting conditions and, first of all, allows eye observation during the night. However, due to the camera sensitivity in this spectrum, the obtained images are of lower contrast and contain higher noise level. Therefore they require special preprocessing. An interesting observation is that in this setup, pupil areas reflect NIR illumination which manifests itself as a bright point in the NIR image of an eye. This is caused by the retina reflecting the infra-red

light back to the camera, assuming that the source of light and the camera are aligned. Therefore it is not a surprise that in many works this phenomenon was proposed to facilitate eye detection in NIR images [61][5]. For example in the system presented by Bergasa *et al.* there are two rings of LEDs alternatively frame-by-frame emitting NIR light. Frames lighted with one of the rings result in very light pupil spots, whereas the others do not [5]. Subtracting the two allows easy pupil detection.

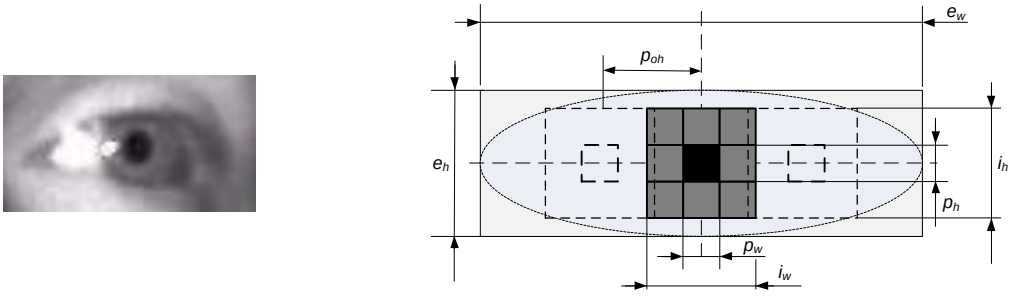


a b

c d

Figure 4. Frames from the slightly over-exposed input NIR images of a driver in night conditions (a,c). Morphologically filtered images (b,d).

However, in our hardware setup we observed that this phenomenon does not lead to reliable detections due to two main reasons. The first one is that to be observed the camera and the LED emitting NIR radiation should be placed close to the axis of sight. Otherwise the point shows and vanishes, depending on the head movements of a driver. The second is that at the required field of view allowing observation of the entire head of a driver, the bright spot of a reflected ray is very small even at resolution of 640x480 pixels. That is, detected areas are only few pixels wide, as shown in Figure 5a. This, considering noise and light reflexes from other objects in the car, causes problems in its reliable detection. Therefore we do not use that feature for eye detection. Instead we observed that good results can be obtained with slightly over-exposed NIR images. In such conditions, especially the pupil, but also the iris areas retain relatively low intensity compared to other face regions. At the same time the iris area is much larger than the mentioned bright spot of the reflected NIR ray. Also, as observed, the reflections of the NIR source in the pupil regions vary with changing conditions in a car, but vary also for the same conditions but with a different driver. Nevertheless, it is also possible that the proposed system can be joined with other eye detection method, for example the one which are based on bright spot detection, for a more reliable operation due to higher diversity.



a b

Figure 5. An example of a human eye (a) and its model for eye detection in the NIR images (b).

Eye detection is based on the proposed human eye model shown in Figure 5b. It was designed to fit specifics of the NIR image processing, as well as to allow fast computations. Based on it the eye detection algorithm was developed, which block structure is shown in Figure 1. The main steps of this algorithm are as follows:

1. Nonlinear filtering;
2. Pupil detection;
3. Iris detection;
4. Eye region hypothesis and verification;

Details of each step of the algorithm are discussed further on in this section. Description of the parameters of the eye model shown in Figure 5b are described in Table 1 and are also discussed in the next part of this section.

Steps of the proposed algorithm for detection of eye regions in the NIR images are presented in Algorithm 1. The first step of the algorithm is to remove noise and the bright spots. In our system this is done by the morphological erosion with an $m_w \times m_h$ square structural element. In our experiments we used $m_w=m_h=3$. Examples of such over-exposed and filtered eye images are shown in Figure 4.

The next step after the filtering in processing of the NIR images is computation of its integral image. This is a data structure which allows fast computation of sums of pixel intensities in any rectangular areas of an image [13]. For this purpose each pixel of an integral image contains partial sums of all pixels from the original image, which positions are above and to the left from a current pixel, as depicted in Figure 6a.

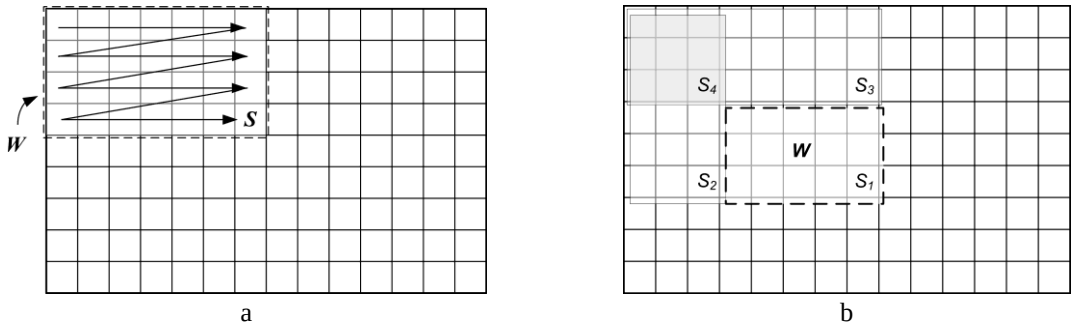


Figure 6. Computation of an integral image (a). Sum of all pixels in a window *W* of any dimensions can be computed with help of the integral image with only two additions and one subtraction (b).

Once the integral image is computed as described, computation of the sum of pixel values within any rectangular window *W* of the original image can be done in linear time equal to reading four values of the integral image and performing two additions and one subtraction, as follows [14]

$$\sum_{\mathbf{x}\in W} I(\mathbf{x}) = (S_1 + S_4) - (S_2 + S_3). \quad \textbf{(1)}$$

where $I(\mathbf{x})$ denotes intensity of a pixel at position $\mathbf{x}=[x,y]^T$ in the original image, while S_1-S_4 are values of the pixels in the corresponding integral image taken at positions presented in Figure 6b. Thanks to this technique sums of pixel intensities in any window can be computed in any window with negligible time penalty.

For the purpose of eye detection the iris-pupil model is proposed as presented in Figure 7. In contrast to some propositions in the literature to define pupil and iris as concentric circles [45], all regions in our model are rectangular. Such approach is justified by the relatively small size of the expected eye regions, small resolution of the NIR images, as well as computational complexity.

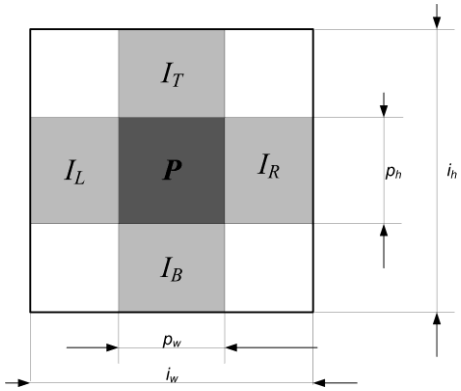


Figure 7. Pixel areas defining pupil and four iris regions taken into computations of their relative average intensities.

As shown in Figure 7, the pupil is represented as a square central region P of dimensions $p_w \times p_h$. Found experimentally values of p_w and p_h as shown in Table 1. These are the reference parameters, since iris and eye dimensions are related to these two. Thanks to this, the number of free parameters which need to be set before the method runs was reduced. In general case such a model has some limitations and does not cover all possible views of the eyes for moving people observed from any direction. However, it is sufficient for driver's observation since in this case the setup camera-driver has strictly limited dimensions. Also, the angle of a driver's sight is assumed to be close to parallel to a direction of a car. The iris is modeled by the four separate rectangular areas I_T, I_B, I_L, I_R, for top, bottom, left, and right iris regions, respectively. Their dimensions are set by dimensions of the pupil rectangle and the i_w and i_h which define width and height of the iris region, respectively (Table 1).

Table 1. Definitions and values of the main parameters of the human eye model for infrared image analysis.

Parameter	**Description**	**Values**
Eye dimensions		
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
Eye detection parameters		
t_p	threshold on a maximal average intensity in a pupil area	128
R_{ip}	range of ratios of average intensity in iris area to average intensity in pupil area	[1.5, 2.75]
$r_{wip} = i_w / p_w$	iris to pupil width ratio	1.9 (±10%)
$r_{hip} = i_h / p_h$	iris to pupil height ratio	1.8 (±15%)
r_{LR}	ratio of average intensity in iris region I_L to I_R	0.2
r_{BT}	ratio of average intensity in iris region I_T to I_B	0.3
r_{TL}	ratio of average intensity in iris region I_L to I_T	0.25
r_{BR}	ratio of average intensity in iris region I_R to I_B	0.25
Other parameters		
L	length of the ordered queue	150
m_w	width of the morphological structural element used for filtering	3
m_h	height of the morphological structural element	3

The pupil candidates are found simply checking an average intensity of each rectangle of size $p_w \times p_h$ in the filtered NIR image to be less than the set threshold t_p (step 4 in Algorithm 1). This threshold is set experimentally, in our experiments we used the value shown in Table 1. Then in the next step of the algorithm, a ratio of an average intensity of an iris area to an average intensity of the pupil area is checked. If it is within the range R_{ip} then the regions is checked further to verify mutual relations between pairs of iris regions, as described in the step 6 of the algorithm. Finally, if all of the aforementioned conditions are fulfilled, then such a pretending region is entered to the priority queue which is ordered by the value of the average pupil intensity σ_P computed in equation (**2**). In other words, the lower the pupil intensity the stronger position in the queue. We tested other comparison measures to control regions in the queue, as well as different lengths of that queue. However, regarding the measure, all of them gave the same results, so σ_P is used since it requires the least computations. On the other hand, the length L of the queue was set to 100, as indicated in Table 1.

The priority queue is depicted in Figure 8. This is a data structure storing pairs: object and a scalar key value. The key value is used to order the objects. In our case the objects describe eye regions, and the ordering value is an average intensity of its pupil region σ_P. If a new object with a key σ_j is to be inserted to the queue, as shown in Figure 8, the action depends on a number of already stored objects in the queue and the value of σ_j. In the simplest case if the queue has some free entries, such a new object is inserted into its position, i.e. to keep ordered values of σ_i. On the other hand, if there are no free entries, then depending on σ_j either the new one is inserted on its position and σ_0 is removed, or the new one is rejected and the state of the queue remains unchanged. Thanks to this we always store up to L of the best pupil positions in the NIR image, and only these are processed in the next steps of the algorithm.

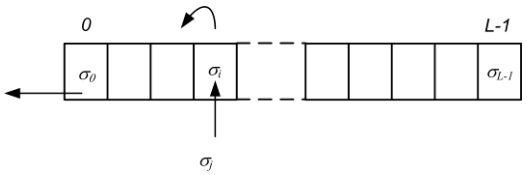


Figure 8. Structure of the priority queue which stores eye region positions and their fit measure σ_P. The queue is ordered in accordance with σ_P. If a new region is inserted to the full queue then the region with the highest σ_P is removed.

1. Process the NIR frame with the morphological erosion filter with the structural element of size $m_w \times m_h$;

2. Compute the corresponding integral image;

3. For each pixel position **do**:

4. If an average intensity in a pupil area P is below the threshold t_p **do**:

$$\sigma_P = \Sigma_{av}\left(P\right) < t_p \tag{2}$$

where $\Sigma_{av}(W)$ denotes computation of an average intensity in a window W;

5. If a ratio of an average intensity of an iris area to an average intensity of the pupil area (Figure 7) should be within the range R_{ip} **do**:

$$\frac{\Sigma_{av}\left(I\right)}{\Sigma_{av}\left(P\right)} \in R_{ip} \tag{3}$$

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

$$\frac{\left|\Sigma_{av}\left(I_a\right) - \Sigma_{av}\left(I_b\right)\right|}{\Sigma_{av}\left(I_a\right) + \Sigma_{av}\left(I_b\right)} < r_{ab} \tag{4}$$

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

7. Insert the region to the queue Q of length L ordered by the values of σ_P computed in (**2**);

8. Refine overlapping eye regions removing the ones with higher value of σ_P.

9. Compute eye regions extending the found pupil-iris regions to dimensions e_w, e_h.

Algorithm 1. The algorithm for detection of eye regions in the NIR images.

Step 8 of Algorithm 1 is to remove overlapping iris regions. If two such regions intersect, the one with the higher average of the pupil area σ_P is removed, thanks to the properties of the queue. The last step consists in computation of eye regions simply dilating iris region to the eye dimensions e_w, e_h, as defined in Table 1. This way found candidates of eye regions are fed to the eye recognition module described in the next section.

The whole model can be adjusted changing parameters contained in Table 1. The good feature of the proposed model is that there are only two input parameters, namely p_w and p_h. All other are relative to these ones. In our experiments good results were obtained for p_w and p_h set as shown in Table 1. However, it is also possible to run simultaneously the eye detector with slightly different values of p_w and p_h. For example, these can be set to 9, 11, 13, and 15 pixels. Then all detections and recognitions are run for these four values concurrently, using one

integral image, however. Nevertheless, such strategy assumes higher computational resources to meet the real-time performance.

4 Eye Detection in the Visible Spectrum

The second processing path presented in Figure 1 is responsible for detection and recognition of driver's eyes in the color images acquired in the visible spectrum. More often than not daily lighting is assumed, although artificial light is also possible. Similarly to the NIR processing path, at first eye candidate regions are detected. These are then refined by the HOSVD classifier, trained this time with prototype patterns corresponding to the daily light conditions. In the following sections the most important modules of the detector are outlined. These were also presented in our report [18]. HOSVD is discussed in section (5).

4.1 Skin Segmentation

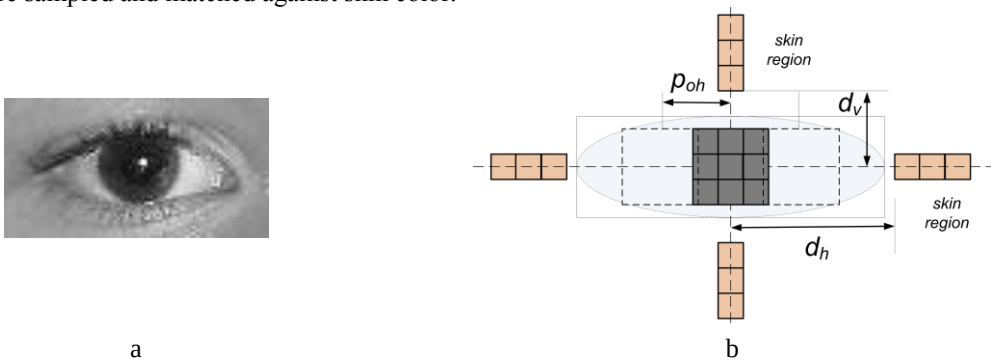
Skin segmentation follows the method proposed by Peer *et al.* [47]. However, an original crisp functions were converted into the fuzzy *IF…THEN* rules to increase robustness, as presented in our previous work [18]. The advantage of this approach is high recall factor and very fast operation. The other advantage of the fuzzy formulation over its crisp version given by Peer *et al.* is that the influence of each particular rule can be controlled separately. Result of each relation is a fuzzy measure. These are then combined with the Mamdani rule. In result, the labeled skin segmentation map is obtained.

Next stage of processing consists in grouping of pixels from the skin segmentation map into compact regions. This is done with the adaptively window growing method (AWG). The main idea is to expand a rectangular window W in all eight direction around a region containing detected skin pixels [14]. The AWG algorithm starts from a single skin point. Then, around such a point a rectangular window is extended in all eight directions, as long as there are sufficient number of skin pixels acquired in each expansion step. In the presented system this showed to be more useful than the method of connected components since the skin points do not need to be connected (sparseness of the points is one of the parameters of the AWG method).

This way obtained candidates for eye regions are selected from the compact skin areas based on the proposed eye model, discussed in the next section. Finally, the eye region candidates which conform to the eye model are fed to the HOSVD classifier.

4.2 Eye Model for Visible Spectrum

Candidates for eye regions are searched in the compact skin regions obtained in the previous step. For this purpose a model has been created (Figure 9b) based on observations of human eyes (Figure 9a). The central part of the model corresponds to an iris of the eye. An assumption is that in real conditions an iris is enclosed by a larger skin region. However, this can be positioned in some distance d_H and d_V, due to images of white of the eye, as well as to the eyebrow. Because of this and to avoid time consuming template matching, only four lines of pixels are sampled and matched against skin color.



a b

Figure 9. An example of a human eye (a). Eye model in normal position for detection of eye region candidates in the skin segmented image (b).

Each match is assessed based on fuzzy rules which allow proper answer on different face positions and lighting conditions. For example, it can happen that due to the viewpoint, an eye is seen only at the border of the face. In such a situation left or right horizontal line in the model (Figure 9b), which corresponds to the skin

region, will not match. Such situations are dealt well by the fuzzy rules which allow also some specific positions of the iris inside the eye white, as shown in Figure 10.

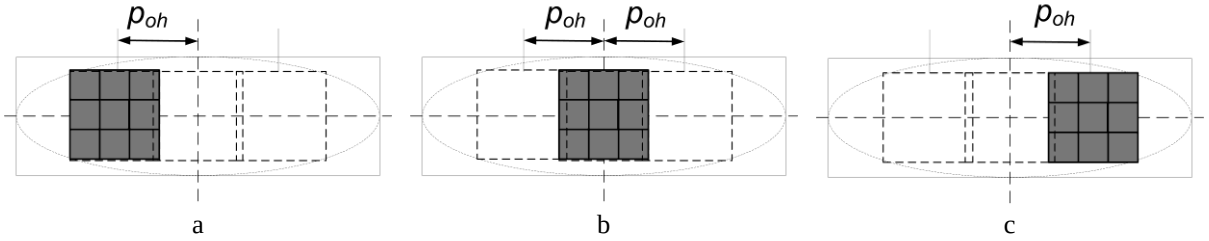


Figure 10. Three distinguished iris positions in our model. The right sight position (a). The central position (b). The left sight position (c). The maximal distance in each direction is set by the p_{oh} parameter.

The parameter p_{oh} controls the allowable horizontal offset of the iris inside the eye region. Its value was defined in Table 1 relatively to the width of the pupil area p_w. The rules in the detection module allow for some false positives which are then sifted out in the second classification module, described in the next section.

5 Eye Recognition in the Space of Deformed Pattern Prototypes

The next module in the cascade of eye classification chain consists of the classifier operating with geometrically deformed eye prototypes. Thus, for each prototype pattern, such as an eye or other object, a tensor is obtained which contains expected views of that pattern. However, to facilitate object recognition we propose two modifications. At first, each prototype pattern, which in our case can belong to the group of eyes and non-eyes objects, is additionally geometrically deformed by small rotations to account for different viewpoints of the camera. Thus, each single prototype is represented as a single 3D tensor containing all these deformations. The second modification is pattern preprocessing with the Euclidean distance transform, which when comparing patterns allows taking into account an influence of all pixels in the patterns, as discussed in (5.1). Thus, each prototype pattern is represented in a form of a deformable pattern tensor. Each such tensor is then decomposed to span a space of orthogonal base tensors. Finally, recognition consists in checking a distance of the projections of the test pattern in all tensor spaces. The closest one indicates a class a test pattern belongs to. Since the prototype patterns are two-dimensional, then a set of them forms a three-dimensional tensor.

Decomposition of the deformable pattern tensors is done with the HOSVD, proposed by de Lathauwer [38]. It is an extension to the well known SVD decomposition of the matrices [25][50]. The method of handwritten digit recognition in orthogonal tensor spaces was proposed by Savas and Eldén [53]. Its version with the tensor of deformable prototype patterns was then proposed by Cyganek to the problem of road signs recognition [15]. In this paper we additionally extend this technique by using the Euclidean distance metrics proposed by Wang *et al.* [59]. All these concepts are briefly described in the next sections.

Mathematical definition of tensors can be found in literature [9][35][38]. In the context of pattern recognition a tensor is considered as a multidimensional array of real values. With this assumption let $\mathcal{T} \in \Re^{N_1 \times N_2 \times \ldots N_P}$ represent a P-dimensional tensor with dimensions counting N_1, N_2, …, N_P. Important information on tensor contents can be obtained by its HOSVD decomposition. Thanks to this any P-dimensional tensor $\mathcal{T}$ can be equivalently represented in the following form

$$\mathcal{T} = \mathcal{Z} \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_P \mathbf{S}_P , \tag{5}$$

where $\mathbf{S}_k$ 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* [38][39]. Let us now observe that for a given mode matrix $\mathbf{S}_i$ equation (5) can be rewritten as follows

$$\mathcal{T} = \sum_{h=1}^{N_P} \mathcal{T}_h \times_P \mathbf{s}_P^h , \tag{6}$$

where $\times_P$ denotes the P-mode tensor matrix product [9][35][38]. In the equation (6)

$$\mathcal{T}_h = \mathcal{Z} \left(\underbrace{:,:,\ldots,:}_{P-1}, h \right) \times_1 \mathbf{S}_1 \times_2 \mathbf{S}_2 \ldots \times_{P-1} \mathbf{S}_{P-1} \tag{7}$$

denote basis tensors and $\mathbf{s}_P^{(h)}$ is an h-th column of the unitary matrix $\mathbf{S}_P$. The index h in (6) and (7) spans the possible values of the P-th dimension, i.e. $1 \le h \le N_P$. Because $\mathcal{T}_h$ is of dimension P-1, then $\times_P$ in (6) denotes an

outer product of two tensors of dimensions P-1 and 1. The result of this operation is a tensor of dimension P, i.e. the same dimension as of tensor $\mathcal{T}$. However, in the context of object recognition the most important feature is that due to the orthogonality of the core tensor $\mathcal{Z}$, tensors $\mathcal{T}_{\mathrm{h}}$ in (6) are also orthogonal. Thus, they span an orthogonal basis of the space defining a given prototype pattern encoded in tensor $\mathcal{T}$.

Based on the above derivations pattern recognition with HOSVD can be stated as testing a distance of a given test pattern $\mathbf{P}_x$ to its projections in each of the spaces spanned by the set of the bases $\mathcal{T}_{\mathrm{h}}$ in (7). This can be expressed as the following minimization problem [53]

$$\min_{c_h} \left\| \mathbf{P}_x - \sum_{h=1}^{N} c_h \mathcal{T}_h \right\|^2 = \min_{c_h} \left\langle \mathbf{P}_x - \sum_{h=1}^{N} c_h \mathcal{T}_h , \mathbf{P}_x - \sum_{h=1}^{N} c_h \mathcal{T}_h \right\rangle, \tag{8}$$

where the scalars c_h denote unknown coordinates of $\mathbf{P}_x$ in the space spanned by $\mathcal{T}_h$, and $N \leq N_P$ denotes a number of chosen dominating components. Higher value of N allows better accuracy, however at an expense of computation time. On the other hand, N cannot exceed N_P which equals the number of deformed patterns in tensor $\mathcal{T}$ [15].

To find a minimum of (8) for each c_h, the set of derivatives with respect to each c_h is computed and then equated to 0. For a tensor of a single class (i.e. one eye example) this leads to the following expression [18]

$$\hat{\rho} = \sum_{h=1}^{N} \left\langle \hat{\mathcal{T}}_h , \hat{\mathbf{P}}_x \right\rangle^2 . \tag{9}$$

As already mentioned, in the case of multiple prototypes for each of them its own deformable tensor is created $\mathcal{T}^k$, where $1 \leq k \leq K$ denotes the class (i.e. a different training object). This means that a test pattern $\mathbf{P}_x$ is checked with all $\hat{\mathcal{T}}_h^k$. In result a series of residue values is obtained, as follows

$$\hat{\rho}_k = \sum_{h=1}^{N_k} \left\langle \hat{\mathcal{T}}_h^k , \hat{\mathbf{P}}_x \right\rangle^2 . \tag{10}$$

Theoretically, N_k in the above can be different for different patterns, but in our experiments all were the same, i.e. $N_k=N$. Finally, for the multi-class classification problem the classifier returns the number of a class k for which its $\hat{\rho}_k$ in (10) is the smallest. In our case if k corresponds to an eye image, then the system reports positive eye recognition. To cope with the problem of outliers, i.e. patterns for which the system was not trained for or the answer is not reliable, a minimal acceptance threshold was set on $\hat{\rho}_i$ (in our experiments it is 0.8).

Finally, in the case of a positive recognition of exactly two eyes there is an additional stage which checks correctness of their geometrical relation. This is done by verifying a set of fuzzy rules, for example checking whether the two regions are not too close or too far apart.

5.1 Pattern Preprocessing with the Euclidean Image Distance

The pattern recognition problem discussed in the previous section and defined in equation (8) assumes the Euclidean metrics for pattern comparison. Given two images $\mathbf{X}, \mathbf{Y} \in \Re^{N \times M}$ their Euclidean distance is defined as follows

$$D_E(\mathbf{X}, \mathbf{Y}) = \sum_{k=1}^{MN} \left(x^k - y^k \right)^2 = (\mathbf{x} - \mathbf{y})^T (\mathbf{x} - \mathbf{y}). \tag{11}$$

where $\mathbf{x}$ and $\mathbf{y}$ are column vectors formed by the column- or row-wise vectorization of the images $\mathbf{X}$ and $\mathbf{Y}$, respectively. In the above the superscript tensor notation is used, i.e. x^k denotes a k-th component of $\mathbf{x}$.

However, the patterns are images which are 2D structures in which a pixel value is as important as its position within the image coordinate space. The second aspect is not easy to be accounted for using the Euclidean metrics exclusively. For this purpose Wang *et al.* proposed a modification, called IMage Euclidean Distance (IMED), which takes into account also spatial relationship among pixels [59]. Let us observe that

Euclidean distance D_E constitutes a special case of a more general distance defined by the metric matrix, given as follows

$$D_G\left(\mathbf{X},\mathbf{Y}\right)=\sum_{k,l=1}^{MN} g_{kl}\left(x^k-y^k\right)\left(x^l-y^l\right)=\left(\mathbf{x}-\mathbf{y}\right)^T\mathbf{G}\left(\mathbf{x}-\mathbf{y}\right). \tag{12}$$

where g_{kl} are elements of the symmetric nonnegative matrix $\mathbf{G}$ of dimensions $MN\times MN$, which defines the metric properties of the space. Wang *et al.* propose to use exponent values for g_{kl} which accounts for pixels positions (apart from their values) in an image. The IMED distance among image $\mathbf{X}$ and $\mathbf{Y}$ is proposed as follows

$$D_{IMED}\left(\mathbf{X},\mathbf{Y}\right)=\frac{1}{2\pi\sigma^2}\sum_{k,l=1}^{MN} e^{-\frac{\left(p_k^1-p_l^1\right)^2+\left(p_k^2-p_l^2\right)^2}{2\sigma^2}}\left(x^k-y^k\right)\left(x^l-y^l\right). \tag{13}$$

where the vector $\mathbf{P}_i=\left[p_i^1,p_i^2\right]^T$ denotes pixel position in the image coordinate system [14]. The D_{IMED} image metric given in (13) is incorporated into the classification algorithms with HOSVD, as described in previous section. This can be achieved substituting D_{IMED} into all places in which the D_E was used.

However, for large datasets of images direct computation of (13) can be expensive. A practical solution to overcome this computational cost was proposed by Sun *et al.* [55] after observing that computation of D_{IMED} can be equivalently stated as a transform domain smoothing. They developed the Convolution Standardized Transform (CST) which approximates well the D_{IMED} . For this purpose the following separable filter was proposed

$$\mathbf{H}=\mathbf{h}\otimes\mathbf{h}^T, \tag{14}$$

where $\otimes$ is the Kronecker product of two vectors $\mathbf{h}$ and $\mathbf{h}^T$. Components of the filter $\mathbf{h}$ were computed by Sun *et al.* as follows

$$\mathbf{h}=\begin{bmatrix}0.0053 & 0.2171 & 0.5519 & 0.2171 & 0.0053\end{bmatrix}^T. \tag{15}$$

The filter $\mathbf{h}$ in (15) was also used in our implementation since it offers much faster computations than direct application of (13). Thanks to the above properties, application of the IMED metrics is straightforward and consists of prefiltering of the input and test patterns with the filter of response defined in equation (15). This is also included as a separate block in the architecture of the system, shown in Figure 1.

6 Experimental Results

For experiments a number of 10-20 minutes long video sequences were recorded in which participated four volunteer drivers. Results of the part of the system operating in the visual spectrum were also presented in our previous paper [18]. Therefore in this paper we mostly discuss results obtained by the NIR vision module.

Figure 11 shows results of the NIR detector operating in images containing persons with open eyes in fronto-parallel position. The three different test sequences obtained in night conditions are shown. The first row in Figure 11 presents all detected pupil regions as computed in the steps 1-7 of Algorithm 1. Middle row shows detected places after the refinement performed in the step 8 of Algorithm 1. Finally, the last row contains candidate eye regions which are fed to the HOSVD eye recognition module. For the more complicated sequences 2 and 3 there is a number of false positives. However, as already mentioned, this is a desired property which increases a rate of true-positives at a cost of some false positives, since the latter will be sifted out by the HOSVD classifier (see Figure 1).

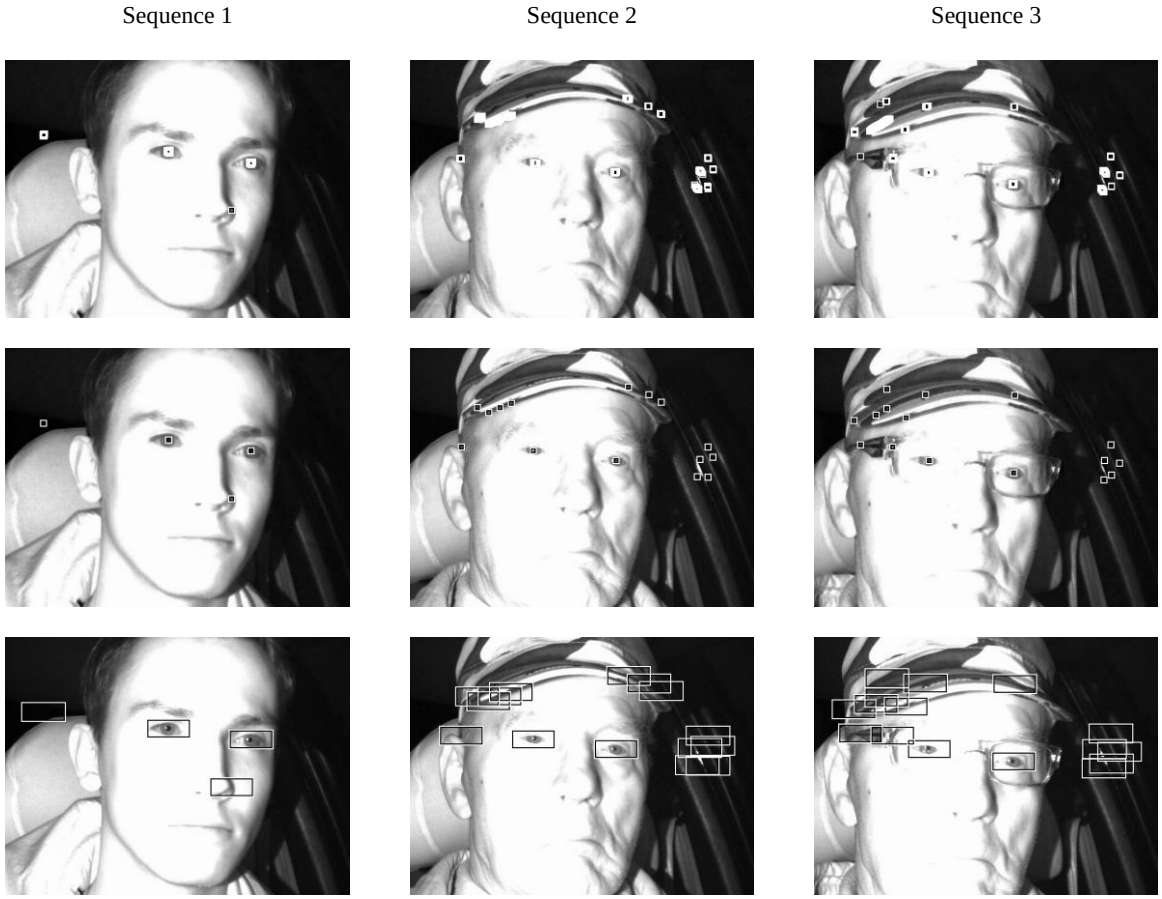


Figure 11. Examples of detection in images showing persons with open eyes in fronto-parallel position for the three test sequences. The first row contains all pupil candidates computed in the steps 1-7 of Algorithm 1. Middle row shows detected places after the refinement performed in the step 8 of Algorithm 1. The last row contains candidate eye regions which are fed to the HOSVD eye recognition module.

Figure 12 depicts results of the NIR detection with for frames with partially visible eyes, usually for the persons in the side view. The same sequences as in Figure 11 were used. The first row presents all detected pupil regions (steps 1-7 of Algorithm 1). Middle row shows detected places after the refinement done in the step 8. The third row contains candidate eye regions which are fed to the HOSVD classifier.

Figure 13 shows examples of detection in images showing persons with closed or not visible eyes for three test sequences. Again, the pupil candidates, these regions after refinement, and candidate eye regions are shown in the consecutive rows in Figure 13. In the second and third sequences there are few false positives which are refined in the HOSVD classifier. It is visible that for the more complicated images, such as a driver with glasses or with hat, there is a number of false positives which are then rejected by the second classification module in the cascade.

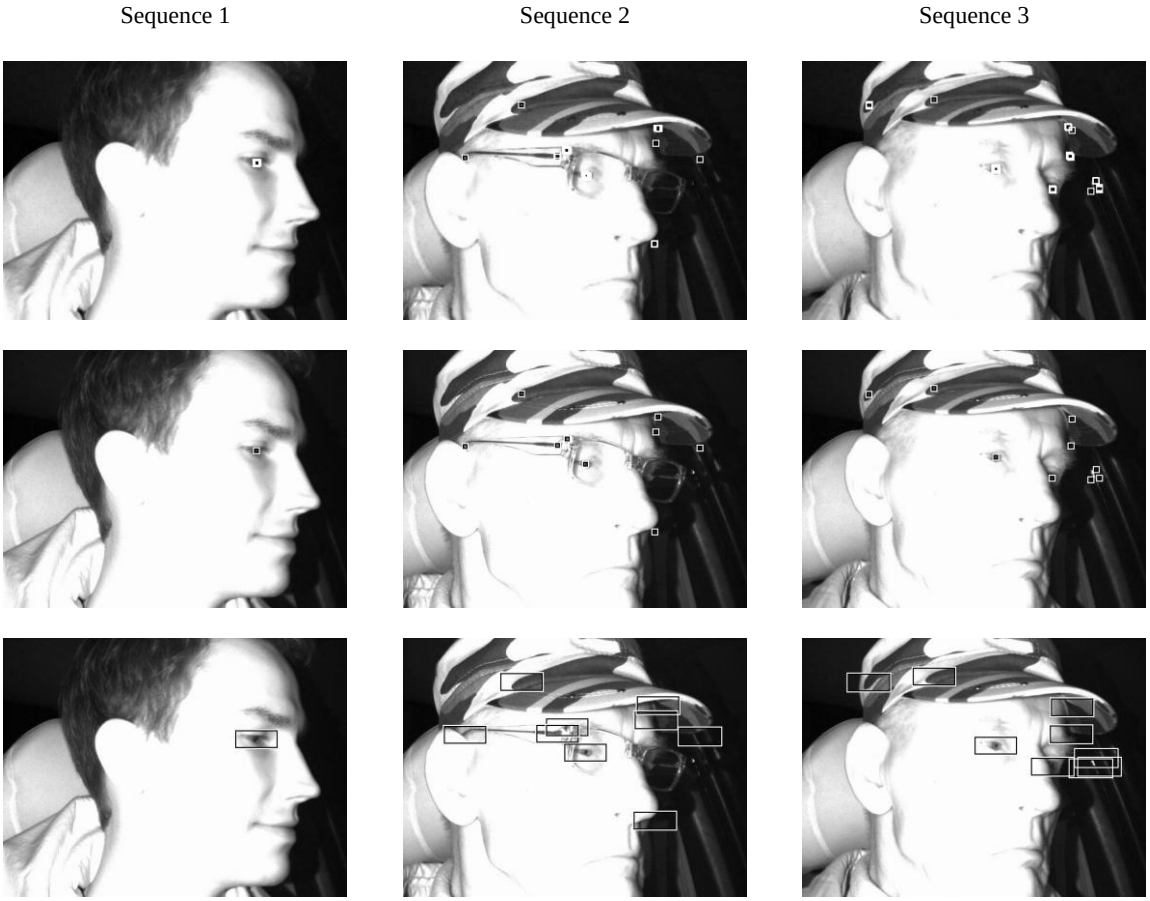


Figure 12. Examples of detections in images with persons with partially opened eyes for the three sequences. The first row shows all pupil candidates computed in the steps 1-7 of Algorithm 1. Middle row shows detected places after the refinement performed in the step 8 of Algorithm 1. The last row contains candidate eye regions which are fed to the HOSVD eye classifier.

As already discussed in section (5) the second in the cascade classification module with the HOSVD classifier requires prior training. For this purpose the two different databases, shown in Figure 14, were created. Figure 14a depicts the database for the classifier in the visible spectrum, whereas Figure 14b for the NIR processing, respectively. Both databases contain two distinct parts: half of the contents for the eye objects, and the second half for non-eye exemplars. The prototypes were cropped from the recorded sequences. However, these are not further used in testing. As already discussed, each prototype object from each of the database is used to build a separate tensor of its deformable versions. An additional use of the IMED metric extends its ability to correctly match the right object during the recognition phase. Thanks to this the number of objects stored in the databases does not need to be very large since the contained exemplars show to be representative for their classes. This, in turn, has an impact on classification time, since a test pattern needs to be checked to each of the tensor spaces associated with each of the prototypes contained in the databases. For this purpose, each such tensor space is decomposed with the HOSVD, as discussed in section (5). In result, the base tensors $\mathcal{T}_h^k$ are computed (k is a class number), which are required during system operation.

Figure 15 depicts first four tensors $\mathcal{T}_h$ and the corresponding core tensors $\mathcal{Z}_n$ for an eye training pattern from the data base in Figure 14a. It is visible that the "energy", defined as a norm of subtensors $\mathcal{Z}_n$, spreads out from the left top corner of the images [38].

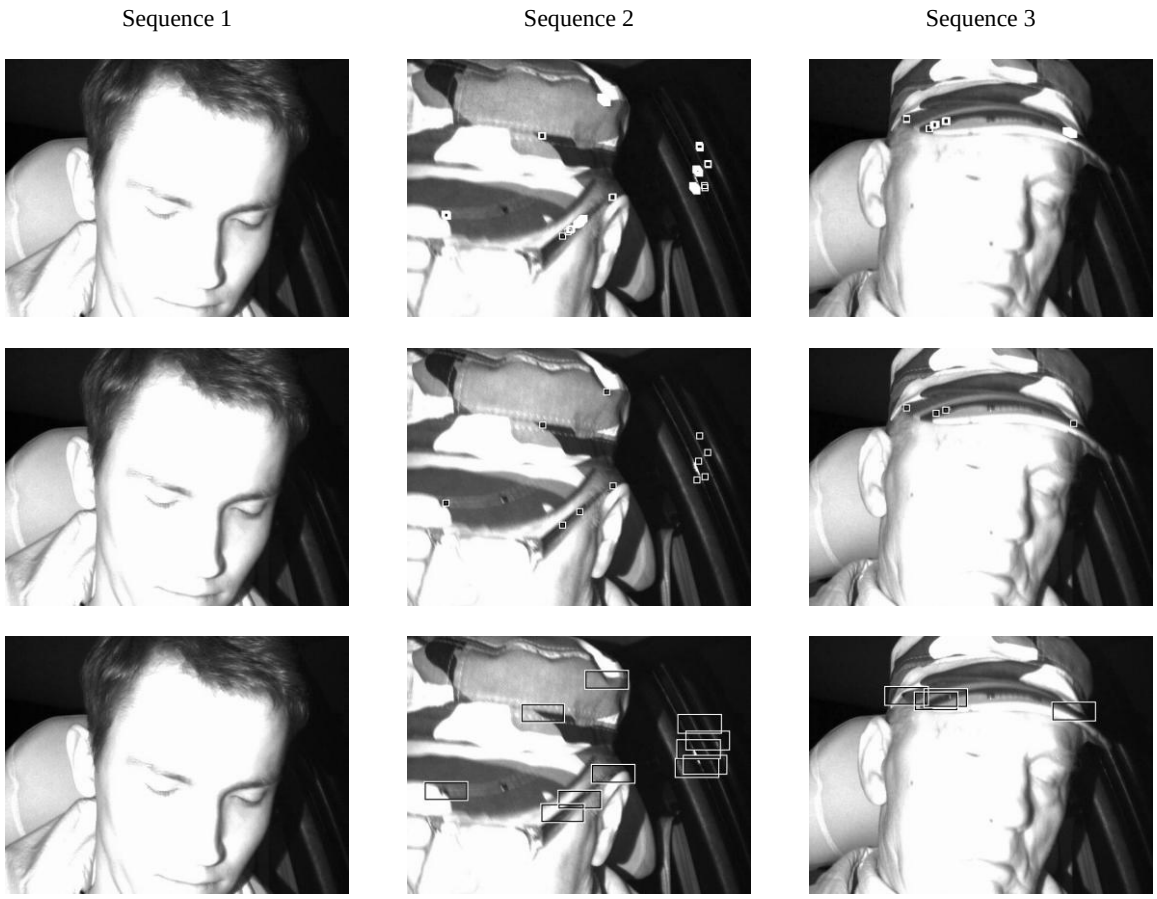


Figure 13. Examples of detection in images with persons with closed or not visible eyes for the three test sequences. The first row shows all pupil candidates computed in the steps 1-7 of Algorithm 1. Middle row shows detected places after the refinement performed in the step 8 of Algorithm 1. The last row contains candidate eye regions. In the second and third sequences there are few false positives which are refined in the HOSVD classifier.

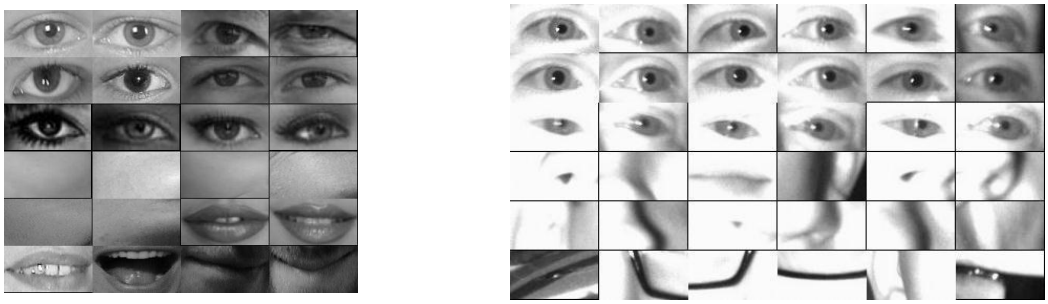


Figure 14. A training database for the HOSVD classifier. A database containing the eye and not-eye patterns for the visible spectrum (a). A database for the NIR spectrum (b).

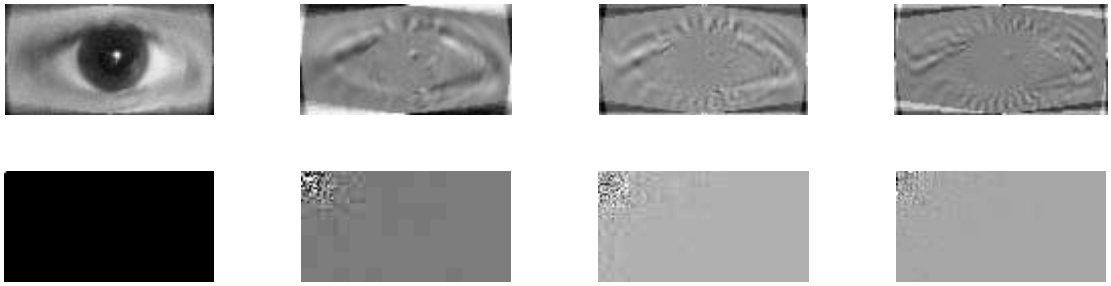


Figure 15. First four tensors $\mathcal{T}_h$ of an eye (first row). Corresponding core tensors $\mathcal{Z}_n$ (second row).

Eye regions in a frame recognized by the HOSVD classifier are further verified to fulfill two conditions:

1. Sufficient response level of the HOSVD classifier;
2. Geometrical relations of the two eyes;

In the case of more positive responses (i.e. if the classifier reports more than two eyes in a frame), only the two with the best responses - i.e. the lowest values of $\hat{\rho}_k$ in (10) - are further process, and the other are simply rejected.

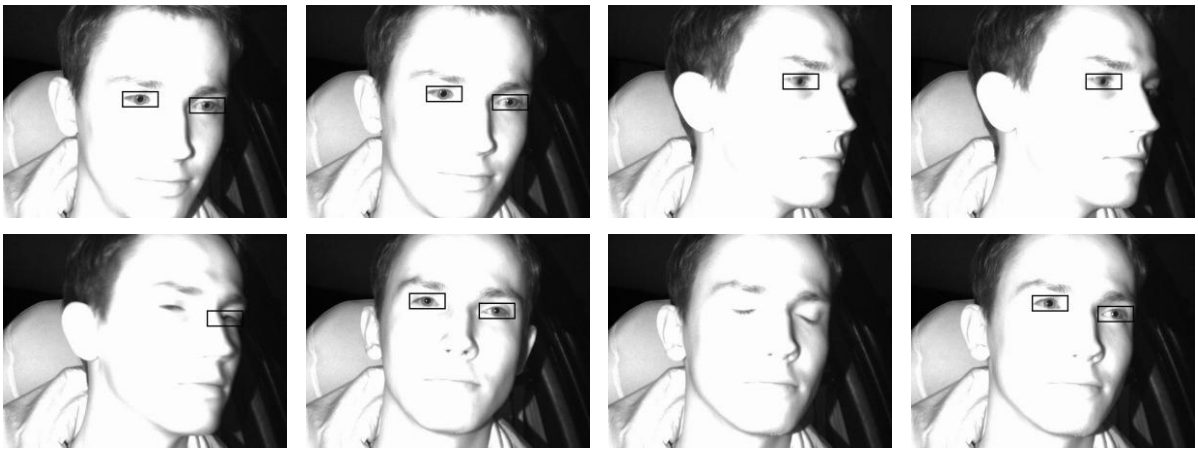


Figure 16. Results of eye recognition in exemplary images taken each tenth frame from sequence 1. Eye regions that passed the HOSVD classifier are shown in black.

Figure 16, Figure 17, and Figure 18 show results of NIR system operation for the three test sequences, respectively. Each shown image is taken at each tenth frame of the sequence. Eye regions that positively passed the HOSVD classifier are shown in black rectangles.

Figure 17. Results of eye recognition in sequence 2. Each consecutive two frames are separated by ten other frames. Eye regions that passed the HOSVD classifier are shown in black.

For detection the parameters from Table 1 were used with initial values of p_w and p_h set to 12 pixels. The HOSVD classifier was trained with the patterns (each 85x47 pixels) from the NIR database, rotated in the range of ±36° with step of 2°. To compute $\hat{\rho}_k$ in (10) N=24 important components were used. Each test pattern before being classified is warped to the size of the patterns used in training, i.e. 85x47 in our case.

Figure 19 shows examples of some false positives returned by the NIR system. In some situations the system is confused by similar patterns to the ones used in HOSVD training. In some respect this is a drawback of the deformable models approach which allows too distant variants of the patterns. We observed that this situation can be alleviated by enlarged patterns stored used to train HOSVD (in our experiments these were in order 80x40), as well as by more examples in the databases. However, the side effect is much higher computation time which does not allow real time operation.

Figure 20 depicts examples of recognitions in laboratory conditions for different positions of head and for a person wearing the glasses. In all cases recognition is facilitated by a slightly overexposed NIR images which

makes skin regions much brighter than the eyes. Thanks to the over-exposure also translucent glasses usually do not deteriorate the positive recognitions.

Figure 18. Results of eye recognition in sequence 3 with a person wearing glasses. Each tenth frame shown. Eye regions that passed the HOSVD classifier are shown in black.

Figure 19. Examples of some false positives from different test sequences.

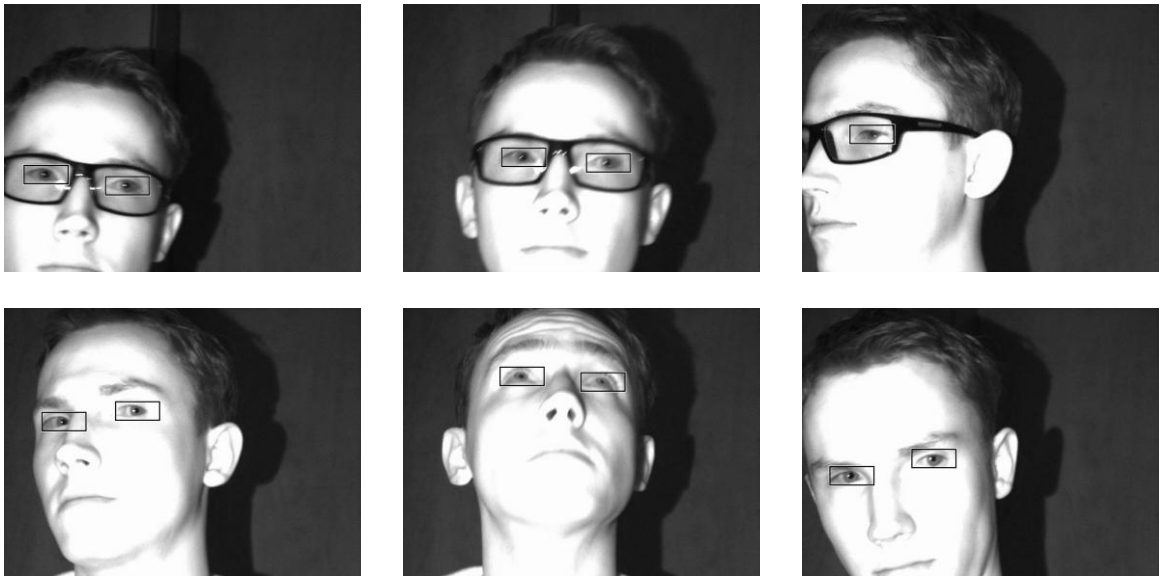


Figure 20. Examples of recognitions in laboratory conditions for different positions of head.

The system in the software implementation allows processing of 24 frames 640x480 per second for the visible and NIR paths. This is sufficient for real-time operation in a moving vehicle endowed with a laptop computer, as shown in Figure 3. However, the proposed algorithms are highly scalable and can be easily ported to a hardware platform if intended to be integrated with other devices of a car and to avoid external computers. This is considered as a further step in development of a complete driver monitoring system which also takes into consideration other sources of information on driver's behavior.

Table 2. Accuracy results obtained for the test NIR sequences shown in Figure 16, Figure 17, and Figure 18.

Sequence no.	**TP [%]**	**FP [%]**	**FN [%]**
1	97.4	0.3	2.3
2	95	0.1	4.9
3	96.5	0.1	3.4
Average			
	96.3	0.17	3.53

Table 2 contains the qualitative parameters of the system for the test sequences for the test sequences which exemplary frames are depicted in Figure 16, Figure 17, and Figure 18. Apart from the true-positive (TP) parameter the second very important is the false-positive (FP). Its level constitutes one of the most important factors of the system since false positives can cause the system not reacting in the cases where driver's eyes are actually closed. On the other hand, the false-negatives mean missed recognitions of the visible and open eyes. However, we observed that the two negative parameters, i.e. FP and FN, are results of errors on separate frames in timeline. In other words, erroneous recognitions occur sparsely and do not tent to group together in neighboring frames of a sequence. This makes them possible to be filtered out in further data processing. Also, the tracking can facilitate to avoid this problem, for example employing the Kalman filter. For instance, in the system presented by Bergasa *et al.* two Kalman filters were used, one for each eye of the driver [5]. However, as shown in their experiments, the Kalman filter often fails to properly predict a position of an eye in frequent situations of abrupt head movements, non frontal orientation of a head, as well as difficult lighting conditions in a car or glasses worn by the drivers. Thus, for this type of the tracker the smooth head movement conditions have to be met. In a case of lost tracking, detection over the whole area of a frame needs to be done. An alternative is to augment the Kalman filter with the mean-shift tracker [10], as proposed by Zhu *et al.* [61]. A discussion of the problems with eye tracking is also provided in the paper by Ahlstrom *et al.* [3]. On the other hand, the presented algorithm is fast enough to perform eye recognition in the frame-by-frame fashion.

Table 3 presents overall average parameters for the tested sequences which were recorded with four volunteer drivers.

Table 3. An average accuracy of the system for all seven tested NIR sequences.

TP [%]	**FP [%]**	**FN [%]**
96.9	0.13	2.97

For a video sequence results of the NIR processing are saved in a meta-data format. That is, for each frame a value from the set {0, 1, 2} is stored, depending simply on a number of positively recognized eyes in that frame.

Figure 21 shows plots of eye activity in the three sequences shown in Figure 16, Figure 17, and Figure 18. Periods of open eyes correspond to level 2. Partially open eyes or conditions with only single eye visible are marked at level 1. Finally, closed eyes periods are correspond to the level of 0. In the simplest scenario and depending on a chosen security level, the periods longer than a second for which the average is less than 1 can turn the alarm. This shows that even the presented system alone can be used as an alerting module. Nevertheless, this is the simplest approach which relies on calculation of the so called percent-of-eye-closure (PERCLOS) parameter to measure driver’s drowsiness which, if used alone, is only in rare conditions leads to the satisfactory operation. Therefore other approaches to the analysis of drivers' behavior are presented in literature which assume a more complex analysis of available information on driver's conditions [5][45]. The presented system was designed as a part of such a more complex monitoring system which uses data from different sensors which provide various types of information on driver's conditions and behavior. On the other hand, as already discussed, our system has some unique features which can make it attractive especially in real car conditions. In this respect especially important are low power NIR illumination and accurate real-time processing.

As already mentioned, in a daily conditions the NIR path of eye recognition can be augmented with the path operating in the visible spectrum, which details were presented in [18]. It relies on human skin segmentation and slightly different eye model which was discussed in section (4). Figure 22 shows results of eye recognition by this module for three frames (top-down). Shown are the input color frames, detected skin areas, then the adaptively grown windows around the binarized skin regions, as well as the recognized eyes from eye candidates.

Experiments were conducted on a database of selected test images containing persons with well visible faces in sufficient lighting conditions, shown in Figure 14a. To test accuracy of the method it was compared with answers of a human operator. The resulting average recall parameter is 95% and precision 98%. This compares favorably with the results reported e.g. in [61][7]. Examination of the misclassified cases reveals that problems are usually due to wrong initial skin segmentation. There are other methods which offer more precise segmentation, although at a cost of higher computational complexity [32][48]. In the context of drivers' monitoring the most promising in this respect is application of the SVM classifiers trained with real skin samples [16]. This is planned for future research.

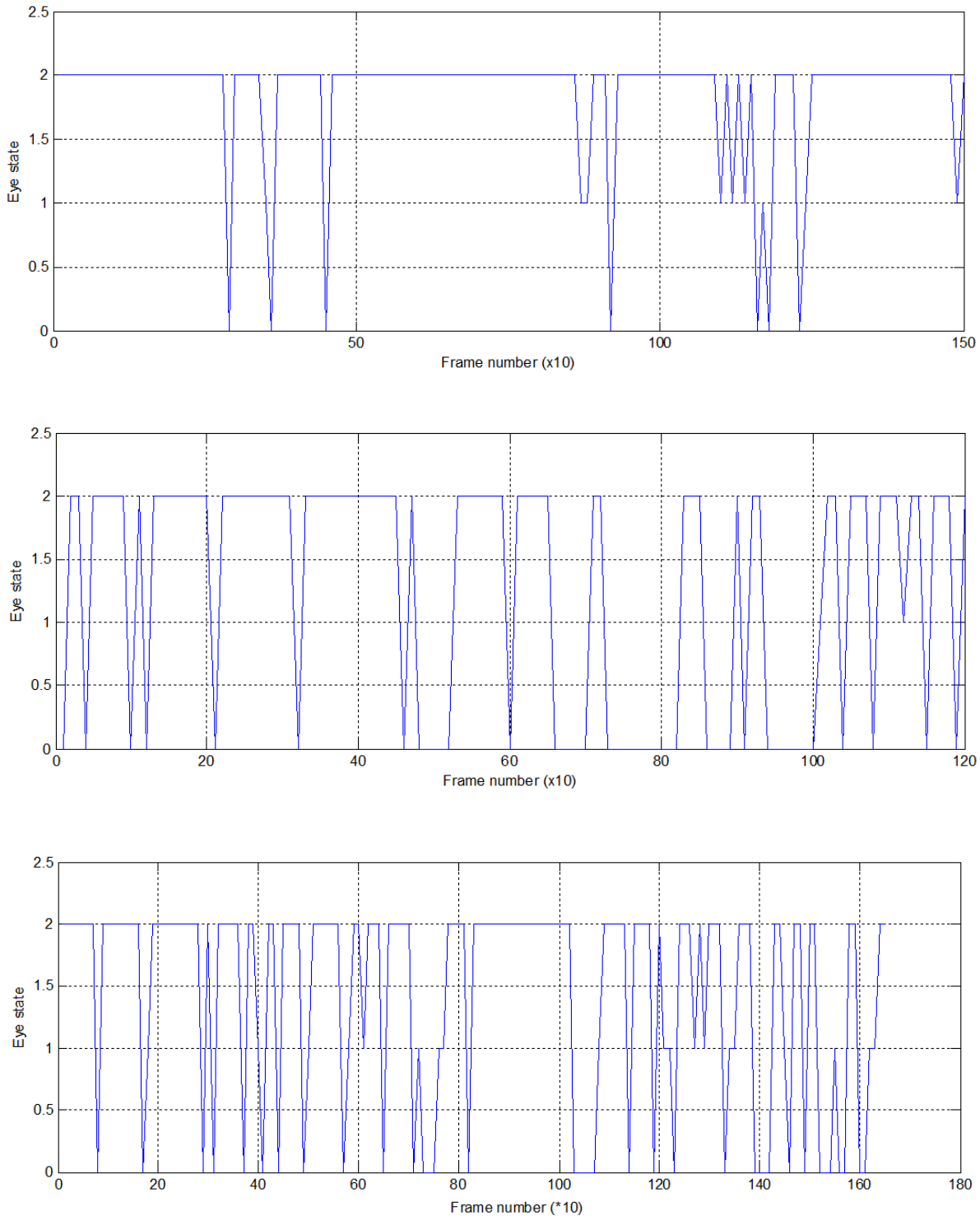


Figure 21. Plots of eye activity in the three sequences shown in Figure 16, Figure 17, and Figure 18. Periods of open eyes shown at level 2. Partially open eyes or conditions with only single eye visible are marked at 1. Closed eyes periods are at level 0.

Figure 22. Results of eye recognition for three frames (top-down) taken with the camera mounted in a car. From left to right: RGB frames, skin binarization maps, human skin regions after the adaptive window growing, recognized eyes from eye candidates.

7 Conclusions

The paper presents a vision system for driver's eye recognition, capable of operating in real car conditions and real-time. The system consists of the two cameras, one color for the visible spectrum, the second one operating in the NIR range. It contains also one NIR illuminating LED mounted on the rail attached to the windshield. Video signals from the cameras are passed with the IEEE 1394b link to the local computer which also runs recognition software which allows real-time processing and reactions on the situations in a car. The paper extends our previous work concerning only processing of the color images which are limited to operation in only daily conditions [18]. Therefore in this paper we focus mostly on the second processing path which operates in the NIR spectrum which is invisible to the driver. However, known NIR-based eye recognition system rely on special effect of white spot in the image resulting from the reflection of the NIR rays by the retina of the eye. To obtain this effect both, the camera and the NIR source, need to be placed central to the axis of sight of the driver. Not always such conditions are easy to be met. Also, movements of a driver cause yaw from the central axis. Therefore, in our system we choose different approach and do not use the white spot effect. Instead, the original vision processing chain, as well as the defined eye model allow for eye detection in any position of the camera. This allows also operation of the system with only single LED with very low power. This is very important factor which affects safety of a driver, since as observed longer exposures to NIR can be tiresome and above certain power level even dangerous.

In both paths of the vision system, the two cascades of classifiers were used. The first of the classifiers is responsible for detection of eye candidates, and the second does validation of these, leaving only recognized eye regions. This twofold strategy allows real-time operation since the first classifier in the chain was design for fast detection which sometimes leaves few false positives. Detection methods in both spectra are different, however. Detection in the color images starts from skin segmentation, followed by the compact skin region extraction with an adaptively window growing method. Then the eye region candidates are selected based on the proposed eye model for color images. Finally, eye validation is done with the HOSVD classifier. On the other hand, detection in the NIR images relies on different eye model designed for this spectrum. In the processing chain, at first the pupil areas are detected, then iris are found and verified. Fast processing in that chain is possible thanks to the used integral image method which allows computation of a sum of pixel values in any rectangle of the image in only four memory reads and three additions. This way found iris are expanded to the eye candidate regions which, after geometrical verification, are fed to the HOSVD classifier, this time trained with eye and non-eye example patterns specific to the NIR conditions. Additionally, the HOSVD classifier is preceded with the module performing transformation of the input patterns to the Image Euclidean Distance. It allows comparison of patterns which takes into consideration two-dimensional distribution of the patterns which usually leads to a higher recognition ratio.

The presented system was tested in real car conditions with help of the four volunteer drivers. Although in this work we focused mostly on processing of visual signals, the output of the recognition allows direct monitoring of condition of a driver by analysis of the PERCLOS parameter. The obtained recognition accuracy

reached 97% with 0.13% of false positives in our test video sequences. The presented method allows driver monitoring in real-time and in various ambient conditions. An exception are drivers wearing dark sunglasses. In such situations it is possible to consider analysis of information coming from other sensors. This is left for further research.

Acknowledgement

The work was financially supported by the National Center for Research and Development under Lider Program, contract no. LIDER/16/66/L-1/09/NCBiR/2010.

References

1. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), pp. 2729–2730, 2009.
2. Agilent Technologies Inc.. Application Note 1118: Compliance of Infrared Communication Products to IEC 825-1 and CENELEC EN 60825-1, 1999.
3. Ahlstrom C., Victor T., Wege C., Steinmetz E.: Processing of Eye/Head-Tracking Data in Large-Scale Naturalistic Driving Data Sets. IEEE Transactions on Intelligent Transportation Systems, Vol. 13, No. 2, pp. 553-564, 2012.
4. AlliedVision: http://www.alliedvisiontec.com/emea/products/cameras/firewire/guppy-pro/f-031bc.html
5. Bergasa L. M., Nuevo J., Sotelo M. A., Barea R., Lopez E.: Visual Monitoring of Driver Inattention, In D. Prokhorov (Ed.): Comput. Intel. in Automotive Applications, SCI 132, pp. 25–51, 2008.
6. Campadelli, P., Lanzarotti, R., Lipori, G.: Eye localization: a survey. In NATO Science Series, 2006.
7. Chiang, C-C., Tai, W-K., Yang, M-T., Huang, Y-T., Huang, C-J.: A novel method for detecting lips, eyes and faces in real time. Real-Time Imaging 9, pp. 277–287, 2003.
8. Chong Y., Chen W., Li Z., Lam W.H.K., Zheng C., Li Q.: Integrated real-time vision-based preceding vehicle detection in urban roads. Neurocomputing (in press), 2012.
9. Cichocki A., Zdunek R., Phan A.H., Amari S-I.: Nonnegative Matrix and Tensor Factorizations. Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. Wiley, 2009.
10. Comaniciu D., Ramesh V., Meer P.: Real-time tracking of non-rigid objects using mean shift. IEEE Conference on Computer Vision and Pattern Recognition CVPR ’00, Vol. 2, pp. 142-149, 2000.
11. Corchado, E., Abraham, A.C., de Carvalho, L.F.: Hybrid intelligent algorithms and applications. Information Science 180(14), pp. 2633–2634, 2010.
12. Corchado E., Graña M., Woźniak M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing, Vol 75, Issue 1, pp. 61-63, 2012.
13. Crow F.C.: Summed-area table for texture mapping. Computer Graphics SIGGRAPH'84, 18(3), pp. 207-212, 1984.
14. Cyganek, B., Siebert, J.P.: An Introduction to 3D Computer Vision Techniques and Algorithms, Wiley, 2009.
15. Cyganek, B.: An Analysis of the Road Signs Classification Based on the Higher-Order Singular Value Decomposition of the Deformable Pattern Tensors, Advanced Concepts for Intelligent Vision Systems Acivs 2010, LNCS 6475, pp. 191–202. Springer, 2010.
16. Cyganek, B.: Image Segmentation with a Hybrid Ensemble of One-Class Support Vector Machines. The International Conference on Hybrid Artificial Intelligence Systems, San Sebastian, Spain, M. Graña Romay et al. (Eds.): HAIS 2010, Part I, LNAI 6076, pp. 256–263. Springer, 2010.
17. Cyganek B.: A Software Framework for Tensor Representation and Decompositions for Object Recognition. Image Processing & Communications, Vol. 15, No. 2, pp. 5-16, 2010.
18. Cyganek B., Gruszczyński S.: Visual System for Drivers' Eye Recognition. The International Conference on Hybrid Artificial Intelligence Systems, E. Corchado, M. Kurzyński, M. Woźniak (Eds.): HAIS 2011, Part I, LNAI 6678, Springer, Heidelberg, pp. 436-443, 2011.
19. Cyganek B.: Adding Parallelism to the Hybrid Image Processing Library in Multi-Threading and Multi-Core Systems. 2nd IEEE International Conference on Networked Embedded Systems for Enterprise Applications (NESEA 2011), Perth, Australia, 2011.
20. Derrac, J., García, S., Herrera, F.: A First Study on the Use of Coevolutionary Algorithms for Instance and Feature Selection. HAIS 2009, Lecture Notes in Computer Science 5572, pp. 557–564. Springer, 2009.
21. Dong Y., Hu Z., Uchimura K., Murayama N.: Driver Inattention Monitoring System for Intelligent Vehicles: A Review. IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 2, pp. 596-614, 2011.
22. Du Y., Hu Q., Ma P., Su X.: Driver Status Recognition by Neighborhood Covering Rules. Lecture Notes on Computer Science, Vol. 6954, pp. 327–336, 2011.
23. García I., Bronte S., Bergasa L. M., Hernandez N., Delgado B., Sevillano M.: Vision-based drowsiness detector for a Realistic Driving Simulator. 13th International IEEE. Annual Conference on Intelligent Transportation Systems Madeira Island, Portugal, 2010.
24. García I., Bronte S., Bergasa L. M., Almazán J., Yebes J.: Vision-based drowsiness detector for Real Driving Conditions. 2012 Intelligent Vehicles Symposium. Alcalá de Henares, Spain, 2012.
25. Golub G.H., Van Loan C.F.: Matrix Computations. The Johns Hopkins University Press, 3rd edition 1996.
26. Golz M., Sommer D., Trutschel U., Sirois B., Edwards D.: Evaluation of fatigue monitoring technologies. Somnologie, Vol. 14, pp. 187–199, 2010.
27. Hsu, R.-L., Abdel-Mottaleb, M., Jain, A.K.: Face Detection in Color Images. IEEE PAMI, 24(5), pp. 696–707, 2002.

28. http://www.aide-eu.org/press_corner.html
29. http://ec.europa.eu/transport/road_safety/users/professional-drivers/index_en.htm
30. http://heavyroute.fehrl.org/?m=32&id_directory=367
31. http://www.wiley.com/legacy/wileychi/cyganek3dcomputer/material.html
32. Jones, M.J., Rehg, J.M.: Statistical Color Models with Application to Pixel-Level Human Skin Detection. IEEE Int. Conf. Pattern Recognition, Vol. 1, pp. 1056–1059, 2000.
33. Kawaguchi, T., Hidaka, D, Rizon, M: Detection of eyes from human faces by Hough transform and separability filter, Int. Conf. Image Processing, Vol. 1, pp. 49–52, 2000.
34. Kittler, J., Hatef, M., Duing, R.P.W., Matas, J.: On Combining Classifiers. IEEE PAMI, Vol. 20, No. 3, pp. 226-239, 1998.
35. Kolda T.G., Bader B.W.: Tensor Decompositions and Applications. SIAM Review, 2008.
36. Kothari, R, Mitchell, J.L.: Detection of eye locations in unconstrained visual images. Int. Conf. Image Processing, Vol. 3, pp. 519–522, 1996.
37. Kuncheva, L.I.: Combining Pattern Classifiers. Wiley, 2004.
38. Lathauwer, de L.: Signal Processing Based on Multilinear Algebra. PhD dissertation, Katholieke Universiteit Leuven 1997.
39. Lathauwer, de L., Moor de, B., Vandewalle, J.: A Multilinear Singular Value Decomposition. SIAM J. Matrix Analysis and App., Vol. 21, No. 4, pp. 1253–1278, 2000.
40. Lee B-G., Chung W-Y.: Driver Alertness Monitoring Using Fusion of Facial Features and Bio-Signals. IEEE Sensors Journal, Vol. 12, No. 7, pp. 2416-2422, 2012.
41. Lee S.J., Jo J., Jung H.G., Park K.R., Kim J.: Real-Time Gaze Estimator Based on Driver's Head Orientation for Forward Collision Warning System. IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 1, pp. 254-267, 2011.
42. Lee W.O., Lee E.C., Park K.R: Blink detection robust to various facial poses. Journal of Neuroscience Methods 193, pp. 356–372, 2010.
43. Lenskiy A.A., Lee J-S.: Driver's Eye Blinking Detection Using Novel Color and Texture Segmentation Algorithms. International Journal of Control, Automation, and Systems, Vol. 10, No. 2, pp. 317-327, 2012.
44. Ma, Y., Ding, X., Wang, Z., Wang, N.: Robust precise eye location under probabilistic framework. IEEE Int. Conf. on Automatic Face and Gesture Recogn. pp. 339–344, 2004.
45. D'Orazio, T., Leo, M., Guaragnella, C., Distante, A.: A visual approach for driver inattention detection. Pattern Recognition 40, pp. 2341–2355, 2007.
46. Pedrycz W., Aliev R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing, Vol 73, Issues 1-3, pp. 10-23, 2009.
47. Peer, P., Kovac, J., Solina, F.: Human skin colour clustering for face detection. EUROCON 2003 – International Conference on Computer as a Tool, 2003.
48. Phung, S.L., Bouzerdoum, A., Chai, D.: Skin Segmentation Using Color Pixel Classification: Analysis and Comparison. IEEE PAMI Vol. 27, No. 1, pp. 148–154, 2005.
49. Polikar, R.: Ensemble Based Systems in Decision Making. IEEE Circuits and Systems Magazine, pp. 21–45 (2006)
50. Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.: Numerical Recipes in C. The Art of Scientific Computing. Third Edition. Cambridge University Press, 2007.
51. Razmpa E., Niat K.S., Saedi B.: Urban Bus Drivers' Sleep Problems and Crash Accidents. Indian Journal Otolaryngol Head Neck Surg, Vol. 63, No. 3, pp. 269–273, 2011.
52. Rosipal R., Peters B., Kecklund G., Åkerstedt T., Gruber G., Woertz M., Anderer P., Dorffner G.: EEG-based drivers' drowsiness monitoring using a hierarchical Gaussian mixture model. HCII2007 - Augmented Cognition, 2007.
53. Savas, B., Eldén, L.: Handwritten digit classification using higher order singular value decomposition. Pattern Recognition, Vol. 40, pp. 993–1003, 2007.
54. Senaratne R., Jap B., Lal S., Hsu A., Halgamuge S., Fischer P.: Comparing two video-based techniques for driver fatigue detection: classification versus optical flow approach. Machine Vision and Applications No. 22, pp. 597–618, 2011.
55. Sun, B., Feng, J.: A Fast Algorithm for Image Euclidean Distance. Chinese Conference on Pattern Recognition CCPR '08, 1-5, 2008.
56. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: TensorFaces. European Conference on Computer Vision, Denmark, pp. 447–460, 2002.
57. Viola P., Jones M.J.: Robust real-time face detection. International Journal of Computer Vision, Vol. 57, No. 2, pp. 137–154, 2004.
58. Wang, P., Green M., Ji, Q., Wayman J.: Automatic Eye Detection and Its Validation. CVPR '05 Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 03, pp. 164–171, 2003.
59. Wang L., Zhang Y., Feng J.: On the Euclidean Distances of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, pp. 1334-1339, 2005.
60. Wozniak, M., Zmyslony, M.: Designing Fusers on the Basis of Discriminants - Evolutionary and Neural Methods of Training. The International Conference on Hybrid Artificial Intelligence Systems, San Sebastian, Spain, M. Graña Romay et al. (Eds.): HAIS 2010, Part I, LNAI 6076, pp. 590–597, 2010.
61. Zhu, Z., Ji, Q.: Robust real-time eye detection and tracking under variable lighting conditions and various face orientations. Computer Vision and Image Understanding 98, pp. 124–154, 2005.
62. Zhou, Z.H., Geng, X.: Projection functions for eye detection. Pattern Recognition 37 (5), pp. 1049–1056, 2004.