# IMAGE ACQUISITION AND PROCESSING FOR MONITORING DRIVER VIGILANCE

Karel Horak, Ilona Kalova

Brno University of Technology
Department of Control and Instrumentation
Kolejni 4, Brno
Czech Republic
{horakk, kalova}@feec.vutbr.cz

Abstract: *The machine vision system intended for estimating driver vigilance is introduced in the paper. Technique of an image acquisition without any additional illuminator and with autonomous exposure algorithm is proposed at the very beginning of the paper. In the second place an algorithm for face detection in color images is introduced as well as algorithm for eyes tracking. Finally, the basic method for driver vigilance estimation is presented in the end of the paper.*

Keywords: *image processing, driver vigilance, eye tracking*

## 1 Introduction

Camera-based monitoring applications definitely create one of the most promising fields in autonomous systems. Along with the growth of road transport, the interest in vision-based safety systems has increased proportionally. Generally, such safety system can be located either outside or inside a car depending on its type. Both the vision-based system for monitoring traffic density and vision-based system for measuring average speed of the car at the certain road section are examples of outside monitoring systems. Vice versa both the embedded vision-based system for detecting accidents and vision-based system for monitoring driver vigilance in real-time are good examples of safety systems inside the car. The main contribution of this paper is to provide an exemplary design and implementation of the last mentioned monitoring system i.e. system for monitoring driver vigilance [2].

The design of a suggested vision-based system for monitoring driver vigilance can be broken down into the three main subsequent modules. The first module represents an image acquisition task. Uniform images of both the satisfactory resolution and frame rate have to be obtained in this module. At the same time, image acquisition platform has to be simple as soon as possible due to a high usability.

The second module represents a driver's eyes localization task. It consists of face segmentation technique and subsequent eyes detection algorithm. Accuracy of eyes coordinates determination is fundamental for the last module. Finally, the third module represents a task of estimation driver vigilance. The current vigilance is estimated on the basis of several driver blinking parameters. Note that it is very important to distinguish natural blinks from hazardous falling asleep usually caused by a fatigue. An overall image processing flow diagram can be seen on the Figure 1.

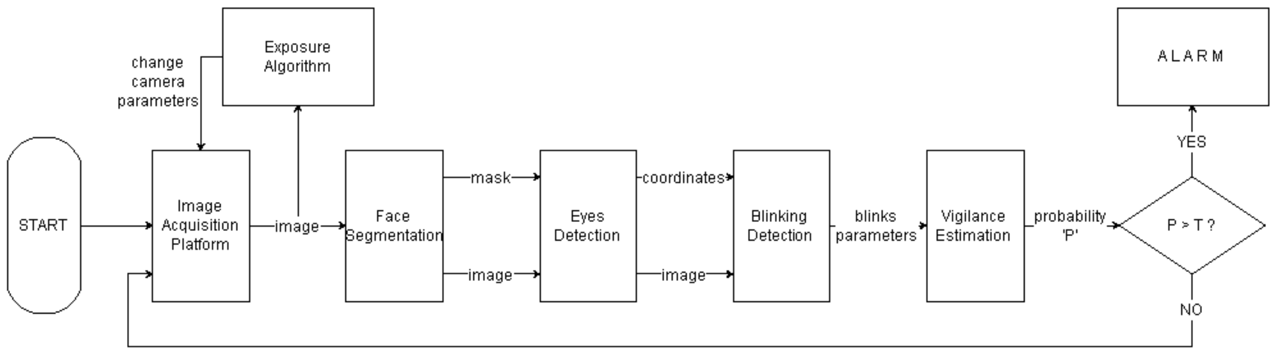


Fig. 1. Flow diagram of the system for monitoring driver vigilance

## 2 Image Acquisition

Generally the first step in almost all machine vision tasks is getting images. Image acquisition for monitoring driver vigilance consists of two separated blocks. First block is an image acquisition platform (i.e. hardware device) described in the next subsection. Second block is a method ensuring uniformity of acquired images under the different light conditions. Proposed ad hoc method for an exposure control is closely described below in the second subsection.

### 2.1 Image Acquisition Platform

The board camera MT9P031 by Aptina Imaging was chosen as a convenient device for an image acquisition task as specified above. The camera is equipped with a CMOS sensor of physical resolution 2592 by 1944 px and 1/2.5″ optical format as you can see on the Figure 2.

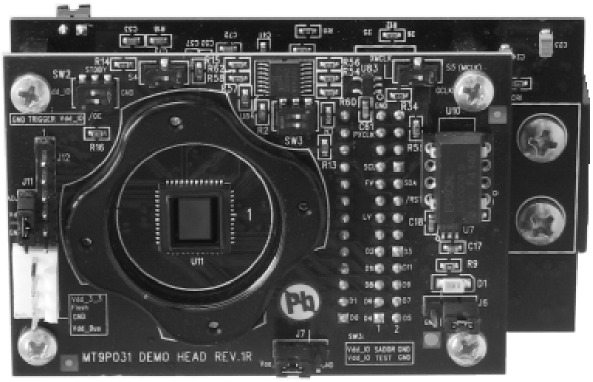


Fig. 2. Board camera MT9P031 by Aptina Imaging

Although physical resolution of the CMOS sensor is relatively high (5 Mpx), the resolution used for an image analysis is only 800 by 600 px because of higher fps provided by the sensor (frames per second). The camera is connected to a computer via USB 2.0 interface. In accordance to the USB 2.0 standard specification the theoretical transfer speed is 480 Mbps. Now consider images of relative low resolution 800 by 600 px and 24 bpp color depth (bits per pixel). In this sense, one image from the camera exactly corresponds to the 11.52 Mb in a raw data format (800x600x24). Then the theoretical number of transferred frames per second is approximately 42 (480 Mbps divided by 11.52 Mb). On the other hand we have only 22.5 acquired images (frames) per second due to the selected CMOS sensor. It only means 54% utilization of USB 2.0 interface. In the case of higher transfer speeds, the USB 3.0 specification with bandwidth of up to 4.8 Gbps can be simply exploited. So high transfer speed makes possible to transfer up to 417 frames per second as specified above.

### 2.2 Exposure Algorithm

The image processing methods generally work better on a series of uniform images. To ensure the uniform or at least similar brightness levels in all acquired images, the following exposure method based on TTL metering was designed (through-the-lens). Note that only predefined image region is considered for a computing of an exposure correction value. A location of such predefined region in an under-exposed image is shown on the Figure 3 (note all images presented here are color in fact and conversion to grayscale is due to paper format).

Fig. 3. Under-exposed image with marked region of an exposure correction

The white rectangle defines an image data for further analysis of exposure correction. A basic idea of all exposure correction methods is simple. An exposure value has to be increased in case of under-exposed image and decreased in case of over-exposed image to make sure that a next image will appear lighter or darker respectively. Exposure correction always represents some type of a control task of the closed loop theory, most often a task with a proportional controller. It follows we need some measured value and some desired value. The measured exposure value ($EV_{ACT}$) is computed from each acquired image in a real time as described below. On the contrary the desired exposure value

($EV_{OPT}$) is empirically predetermined value in the view of the correct function of the following image processing methods. For example both the measured and desired value can be simply represented by the means (or weighted means) of image histograms.
In our case of image acquisition of a driver inside the car an overall image intensity can vary significantly. In the same time the application needs clean and well lighted images in order to optimally perform what it is made for. As mentioned above the desired exposure value is predetermined from a set of test images. $EV_{OPT}$ is exactly equal to an average brightness level of pixels in the selected region. On the contrary the measured exposure value is computed for each acquired image repeatedly in the sense of equation (1).

$$EV_{ACT} = \frac{1}{size(ROI)} \cdot \sum_{(x,y)\in ROI} f(x,y) \tag{1}$$

Only some pixels from ROI can be used for $EV_{ACT}$ computing if real-time processing is a crucial factor of an application. Similarly individual pixels values can be variously weighted e.g. in the direction of the region center.
Based on the difference between $EV_{OPT}$ and $EV_{ACT}$ an exposure correction is performed. It is very important to say, that we assume linearity of an image sensor. It means doubling in the amount of incident light corresponds to doubling in the exposure value. Such assumption is comparatively accurate with respect to determining of both the measured and desired exposure value. Especially a method of determining $EV_{OPT}$ is very dependent on a solved application.
Up to the present we defined both the measured and desired exposure value as a function of brightness levels in the ROI. Here we have to modify an exposure time of the image sensor in order to take an effect. The first frame in an image sequence is naturally exposed with a default exposure time. After that, the exposure time (ET) is modified on the basis of difference between both measured and desired exposure values. The new exposure time value is derived from the equation (2).

$$ET^{(k)} = ET^{(k-1)} \cdot \frac{EV_{OPT}}{EV_{ACT}} \tag{2}$$

Alternative computation of the new exposure time ($ET^{(k)}$) exploits a logarithmic nonlinearity. This method adds (or subtracts) a logarithmic equation term to the old exposure time value ($ET^{(k-1)}$). Such alternative equation (3) converges more quickly than the previous one, but computational costs are slightly higher because of logarithm evaluating.

$$ET^{(k)} = ET^{(k-1)} + \log_2 \frac{EV_{OPT}}{EV_{ACT}} \tag{3}$$

The corrected exposure time value is then written into the appropriate sensor register and takes an effect on the next acquired image immediately. Note that because of a stable output of the auto-exposure algorithm, a hysteresis is used on the difference between $EV_{OPT}$ and $EV_{ACT}$. It means there will only be a modification of the exposure time if the currently computed difference between $EV_{OPT}$ and $EV_{ACT}$ is larger than a certain specific value. This prevents from alternating of exposure time owing to insignificant changes in the acquired scene.
Example of an image acquired with implemented auto-exposure technique is shown on the Figure 4. You can see that such image is much more comfortable to the next image processing methods.

Fig. 4. Well-exposed image with marked region of an exposure correction

A subjective perception of an image quality can be easily compared with more precise image quality indicator. The correct function of the following image processing methods (i.e. face segmentation and eyes tracking) is conditioned by the presence of almost all brightness levels in the acquired image histogram. On the Figure 5 ROIs histograms of both the under-exposed and well-exposed image are shown for comparison. It is obvious that well-exposed image contains more details in more brightness levels.

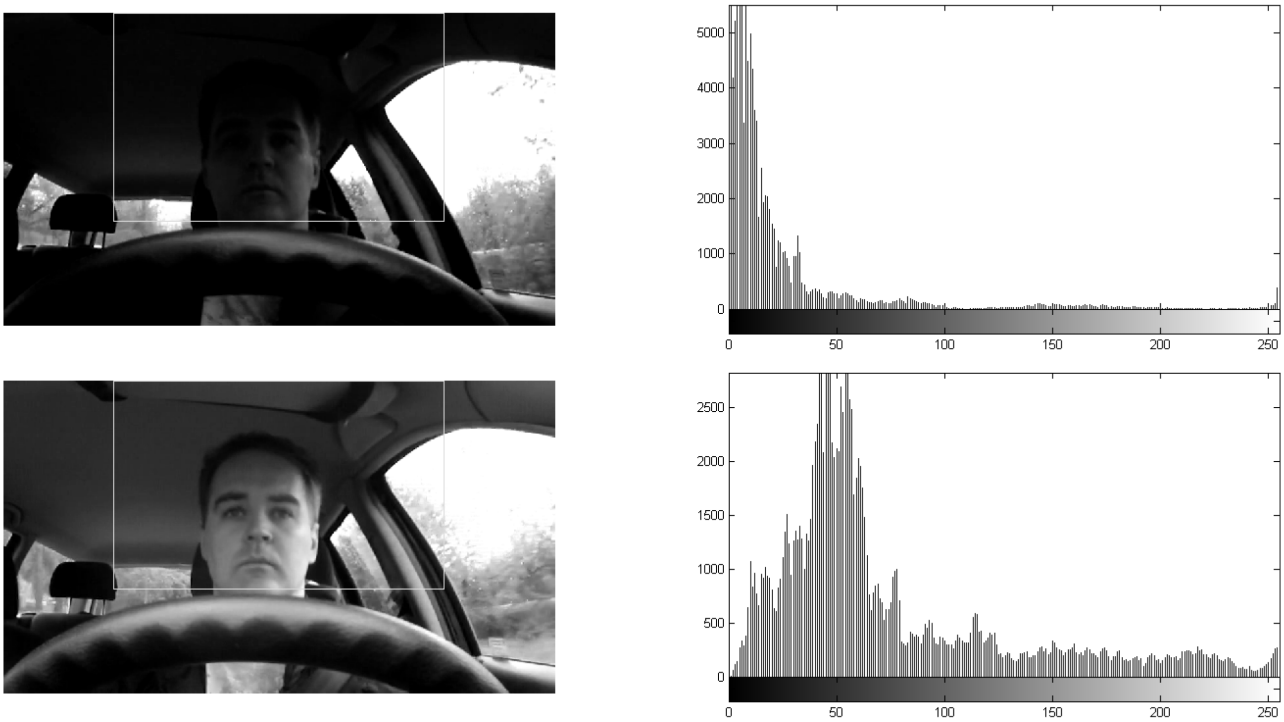


Fig. 5. Under-exposed (at the top) and well-exposed (at the bottom) images with corresponding histograms

Because of the above proposed auto-exposure technique there are no additional lighting requirements or other special devices needed for the correct function of the monitoring system. Such design of the vision-based monitoring system simplifies its using for all individuals significantly.

## 3 Face Segmentation

Human skin detection is most often the first step in applications as face recognition, facial features detection, video surveillance etc. [3]. Here a driver face is localized simply by means of the color segmentation technique (as mentioned above all images shown here are in grayscale due to paper format). The meaning of such localization is only reduction of the ROI due to both the subsequent image processing and overall time costs.

The almost all color segmentation methods are based on a previously generated model of human skin in a certain color space. An arbitrary color space with separated chromatic components (e.g. the YCbCr color space) is convenient for such face segmentation. Each image pixel outside the representative hyperspace in chosen color space is marked as non-face pixel and each image pixel inside the hyperspace is marked as face pixel (see Figure 6). Either the two-dimensional ellipse or three-dimensional ellipsoid is most often used as a representative hyperspace in color space. Dimensionality of hyperspace depends on using (YCbCr space) or not using (CbCr space) a brightness component in the model.

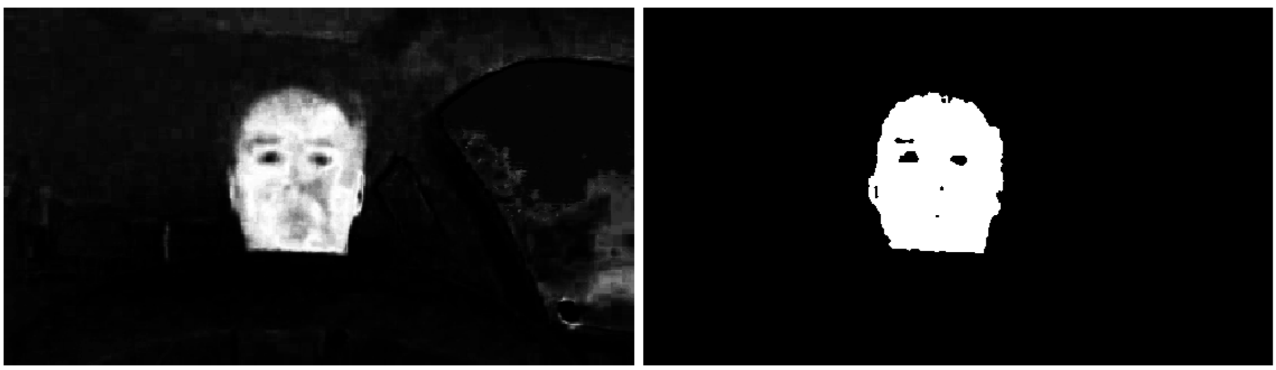

Fig. 6. Results of color-based face segmentation a) probabilistic b) deterministic

Face segmentation is processed on the whole acquired frame by contrast to auto-exposure algorithm, which only works on a predetermined ROI. It means not only face region with a skin color can appear in the segmented image. In the case of more than one compact area marked as a face region, only the largest center-weighted area is marked as a true face. Other passengers in the car, the driver's hands on steering wheel or other driver's body parts can cause additional areas as you can see on the Figure 7.

Fig. 7. Choosing the true face region in the probabilistic image

When the true face region is localized within the acquired image, both driver eyes can be detected and tracked inside this region in order to later estimation of driver vigilance.

## 4 Eyes Detection and Tracking

It is clear that driver eyes inside the car while driving are necessarily at the top part of driver's face. Therefore only an upper half of previously detected face region is used for eyes detection and tracking algorithm and a lower half is then removed from the further image analysis (Figure 8).

Fig. 8. Face segmentation and ROI reduction for eyes detection algorithm

Owing to robust image acquisition and image processing we have only small image region where driver eyes have to be detected. There are a lot of research papers dealing with eyes detection methods in the computer vision field. These eyes detection methods can be clearly divided into two categories, active and passive. Active methods exploit some kind of special illuminator, most often with infra-red wavelengths. It allows detecting pupils more comfortably because of strong reflection infra-rays from an eye ground. Vice versa passive methods do not use any additional illuminator and utilize only ambient light. Within the framework of passive methods a lot of features are extracted from images due to eyes localization. For example templates, image gradients, wavelets, Haar's features, projections, Gabor wavelets are most often used features [4].

In our application we derived a simple eyes detector based on a variance filter and cross correlation. During experiments we discover that eyes regions have higher variance than surrounding regions. This feature not has to be obvious at first glance, but every eye region contains a lot of image gradients because of pupil, sclera and eyelash [1]. So variance is a basic statistical quality and describes the diversity of a random variable. In the image domain it is the second-order moment indicating the variation of gray intensities in the image. If term *f(x,y)* denotes an image, variance is defined as follows.

$$\sigma_{\Omega}^{2} = \frac{1}{\|\Omega\|} \cdot \sum_{(x,y)\in\Omega} \left( f(x,y) - \mu_f(x,y) \right)^2 \qquad (4)$$

where $\Omega$ and $\mu_f(x,y)$ represent area and average gray level on the domain $\Omega$ respectively. Term ||x|| denotes a size of argument i.e. overall number of pixels on the domain $\Omega$. It is clear that variance is rotational invariant and depends on gray level changes only [1]. In our case the image variance $f_V(m,n)$ is always computed on a small part of the face image. These small image blocks are given by their nearest neighborhood denoted $\Omega$. Image variance is then defined as:

$$f_V(m,n) = \sigma_\Omega^2 \qquad where \qquad \Omega = \{(m-1)\cdot k < x < (m+1)\cdot k, (n-1)\cdot k < y < (n+1)\cdot k\} \tag{5}$$

where k denotes size of neighborhood to compute variance. On the next figure is shown resulting image variance as just defined (see Figure 9). The entire face regions (i.e. theirs both the upper and lower half) are shown together in the following figure due to better perception. Note that only upper half of the face region is considered for proposed eyes detection and tracking algorithm.

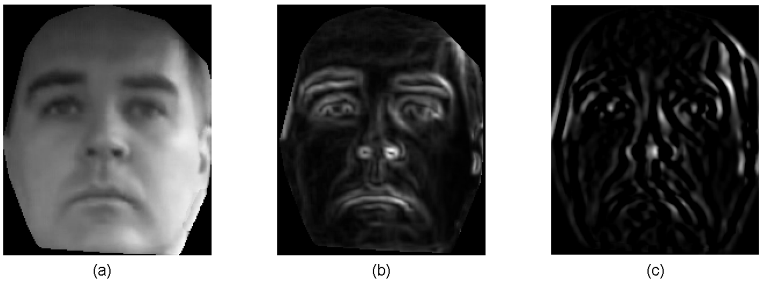


Fig. 9. Segmented face region (a), variance image (b) and cross correlation image (c)

On the basis of previously generated template of an eye variance image, it is possible to perform a cross correlation. The correlation is evaluated over current variance image and stored eyes variance template. It results in grayscale image like depicted in Figure 9c. It is clear that singular points detection is final step in eyes tracking algorithm.

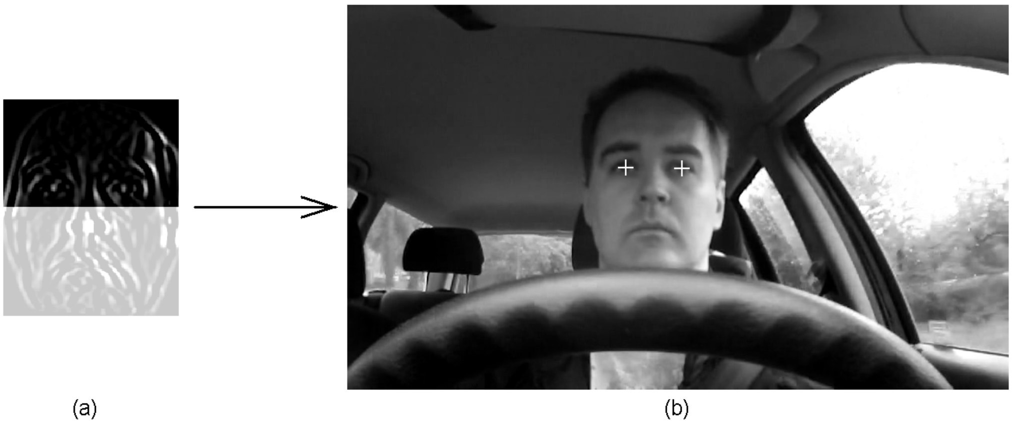


Fig. 10. ROI in correlation image (a) and resulting eyes coordinates (b)

As is shown on the Figure 10 both the *x* and *y* coordinates of driver eyes is localized. Accurate and robust tracking is performed till the driver faces forward. In the case of driver blink the detected eyes points are naturally incorrect. Because of remembered set of previous eyes coordinates, the correct locations are quickly restored when the eyes open.

## 5 Estimation Driver Vigilance

From eyes tracker algorithm we obtain set of x-coordinates and set of y-coordinates in temporal domain. The only first one set is important for a following driver vigilance determination. Physically the distance between centers of human eyes is naturally fixed. It means we can assume the distance between two significant peaks in correlation image (Figure 9c) is time consistent. As well the difference between vertical coordinates of both the left and right eye has to be small in course of time [2].

Driver vigilance is analyzed from x-coordinates waveforms of driver eyes (see Figure 11). As can be seen in the figure, blinking points can be detected when x-coordinates of both the left and right eye changes very significantly. If only one of the two waveforms rapidly changes its value, the detection error is signalized and the correct eye coordinate is maintained owing to previous stored values. This filtering follows to smooth waveforms as depicted in the Figure 12. These smooth waveforms are suitable for the eyes distance verification in temporal domain.

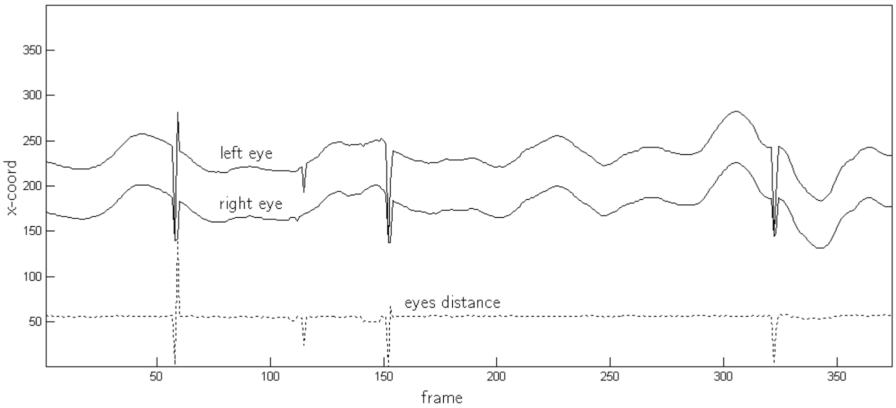

Fig. 11. Horizontal coordinates of detected driver pupils and a distance between them (dotted line)

Generally driver vigilance causes both the more frequent and longer blinking. First of all the mean time is computed from all pairs of the two consequential blinks. Mean time of driver blinking is usually five second approximately. After that the blinking frequency is evaluated as reciprocal value of the blinking mean time.

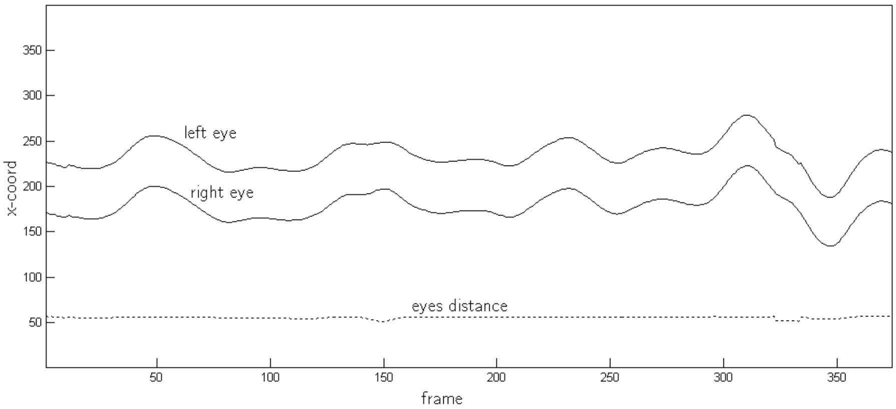

Fig. 12. Filtered (true) horizontal coordinates and a distance between them (dotted line)

When a driver starts his journey a relative vigilance-value is set to 100 percent. If the blinking frequency and/or blinking duration increase for a certain amount of time, this relative vigilance-value decreases proportionally. When a critical value is reached any warning system has to be activated.

## 6 Conclusion

In the presented paper a vision-based system for monitoring driver vigilance was introduced. The robust and efficient image acquisition technique with auto-exposure algorithm was designed in order to achieve good starting point for the next image processing steps. After short description of the face detection method and eyes tracking algorithm the basic preview of estimation driver vigilance was shown.

**Acknowledgement:** The presented paper was supported by the Czech Science Foundation under the project GA102/09/1897 and by the Ministry of Education of the Czech Republic under the project MSM0021630529. Both are very gratefully acknowledged.

## References:

[1] Feng, G.C., Yuen, P.C., Multi-cues eye detection on gray intensity image, *The Journal of the Pattern Recognition Society*, 2001, pp. 1033-1046.
[2] Horak, K., Richter ,M., Kalova, I., Human Eyes Localization for Driver Inattention Monitoring System, *15th International Conference on Soft Computing*, 2009, pp. 283-288.
[3] Viola, P., Jones, M.J., Robust Real-Time Face Detection, *International Journal of Computer Vision*, 2004, pp. 137-154.
[4] Vural, E. et al., Drowsy Driver Detection through Facial Movement Analysis, *Human–Computer Interaction*, Vol. 4796, 2007, pp. 6-18.