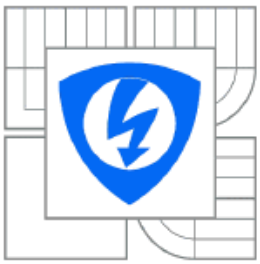
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF CONTROL AND INSTRUMENTATION

EXTRAKCE OBLIČEJOVÝCH ÚNAVOVÝCH CHARAKTERISTIK ŘIDIČE

EXTRACTION OF DRIVER'S FACIAL FATIGUE FEATURES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. PETR KOCICH
AUTHOR

VEDOUCÍ PRÁCE — Ing. KAREL HORÁK, Ph.D.
SUPERVISOR

BRNO 2011

BRNO UNIVERSITY
OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

Department of Control and Instrumentation

Diploma thesis

master's study field
Cybernetics, control and Measurements

Student: Bc. Petr Kocich ***ID:*** 78112
Year of study: 2 ***Academic year:*** 2010/11

TITLE OF THESIS:

Extraction of driver's facial fatigue features

INSTRUCTION:

The main aim of the thesis is to propose and implement algorithms intended for driver's fatigue detection in the course of driving. Algorithms should be based on biological fatigue characteristics of a driver as blinking frequency, blinking time period, blinking base, yawning etc. Thesis output is C/C++ implementation of the robust algorithm for detection of the level driver's fatigue before potential microsleep.

REFERENCE:

[1] VERNON, David. Machine Vision : Automated Visual Inspection and Robot Vision. Hemel Hempstead : Prentice Hall International (UK) Ltd., 1991. 260 p. ISBN 0-13-543398-3.
[2] SONKA, Milan, HLAVAC, Vaclav, BOYLE, Roger. Image Processing, Analysis and Machine Vision. 3rd edition. Toronto : Thomson, 2008. 829 p. ISBN 978-0-495-08252-1.
[3] RUSS, J.C. The Image Processing Handbook. Boca Raton : CRC Press, 1995. 674 p. ISBN 0-8493-2516-1.

Assigment deadline: 7.2.2011 ***Submission deadline:*** 23.5.2011

Head of thesis: Ing. Karel Horák, Ph.D.
Consultant:

prof. Ing. Pavel Jura, CSc.
Subject Council chairman

WARNING:

Abstrakt:

V této práci jsou otestovány metody pro detekce hlavy a očí v černobílém a barevném snímku. Je proveden experiment změřena doba a frekvence mrknutí při různých únavových stavech. Navržený detektor hlavy pracuje s rychlostí 10 fps v černobílém snímku.

Klíčová slova:

detekce oka, segmentace tváře, detekce kůže, únava řidiče, zpracování obrazu

Abstract:

In this paper, the methods for head and eyes localization in grayscale and color image are tested and an experiment of blink duration and blink frequency is described. The proposed head detector works at 10 fps using grayscale image.

Key words:

eye detection, face segmentation, skin detection, driver's fatigue, image processing

KOCICH, P. *Extraction of driver's facial fatigue features.* Brno University Of Technology, Faculty of Eletrical Engineering and communication, Department of Control and Instrumentation, 2011. 68 pages. Tutor in the Czech republic Ing. Karel Horák, Ph.D., tutor in Spain Dr. Nestor Arana Arexolaleiba

Prohlášení

„Prohlašuji, že svou diplomovou práci na téma Extrakce obličejových únavových charakteristik řidiče jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavyVI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **23. května 2011** ……

podpis autora

Poděkování

Děkuji vedoucímu diplomové práce Ing. Karlu Horákovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práci.

V Brně dne: **23. května 2011** ……

podpis autora

INDEX

LIST OF PICTURES

1. INTRODUCTION

According to the statistics collected by the police, 900 traffic accidents per year are caused by micro-sleep in the Czech Republic. The percentage of the accidents caused by fatigue varies between 5 – 25 % depending on the individual study [32].

There are a lot of projects studying the fatigue detection, and they are financed by the automotive industry. For example, Mercedes-Benz presented a system called Attention Assist which is based on 70 fatigue features, including facial fatigue features. Nearly all leading manufactures are trying to develop their own system for detecting driver's drowsiness and improve their cars. A number of these systems is listed in the sources section [32]. As shown later [31], such systems may be interesting for professional drivers. Using cameras installed in vehicles might serve other purposes as well -- hands-free and video calls, driver recognition, to name a few. A couple of approaches for driver fatigue monitoring has been developed (these are listed in the sources section [25],[26],[33]).

However, using a camera for detecting driver's drowsiness brings substantial benefits. It is a low price (a camera could be used for extracting different facial fatigue features) and compact solution (a camera + processing unit), it is a driver untouched method, etc. The goal of this project is extraction of driver facial fatigue features with a color camera placed in the vehicle interior. These features include duration of blinks, time between blink events, eye gaze, yawning, nodding [25]. The duration of blinks seems to be one of the most reliable facial fatigue features [20]. We have made a decision that this project will be based on duration of blinks (the reason for this decision is delivered in the section 2.1). Therefore, it is necessary to determine the minimal possible fps for the camera to be used for measuring the time of a blink. After that, a robust head/eye classifier should be developed in order to extract facial fatigue features. The proposed algorithm should be robust enough to work under different daylight conditions and fast enough to capture blink event. Finding a face during night is not the goal of this work. The final product should be able to prevent the driver from falling in a micro-sleep. It is obvious, though, that the

driver will fall in a micro-sleep if the fatigue level is high. The fatigue level detection is the first task to perform if we want to warn the driver prior to falling in a micro-sleep.

The paper is organized as follows: Section 2, fatigue experiment. Section 3 presents different methods for head/eyes segmentation and proposes a new algorithm for head detection. In Section 4, an application in OpenCV is presented. Finally, conclusions, future work and appendixes make up Sections 5, 6 and 7.

2. STATE-OF-THE-ART

2.1 FACE DETECTION

Face detection is a very complex problem in grayscale images. Many scientific papers have been published on this topic [8][9]. One of the most used approaches was published by Viola-Jones [29]. This algorithm uses AdaBoost machine learning algorithm for generating pyramidal structure of face features. Many works showed that this algorithm is very robust [37]. The face features are calculated from Haar wavelets. The Haar wavelets are computed from the Integral image (7.12) which is a cumulative sum of an image. To provide a robust face classification it is important that hundreds face features be calculated. The integral image is used for fast convolution calculation (7.12). This face finding algorithm has alredy been used in smart sensors for cameras.

2.2 FATIGUE DETECTION

A number of papers on facial fatigue features and micro-sleeps events have been studied [18],[22],[23],[24],[25],[26]. In section 14.17, a list of the most widely used and the most reliable facial fatigue features is presented. It has been found that it is necessary to monitor more than one facial fatigue feature to determine a fatigue level. We have decided that, for the purpose of this project, the average eye open/closure time, time of closed eye during blinking, PERCLOS and frequency of blinks will be calculated. All these facial fatigue features can be calculated from one dimensional time-dependent characteristics which represents the degree of eyes closing.

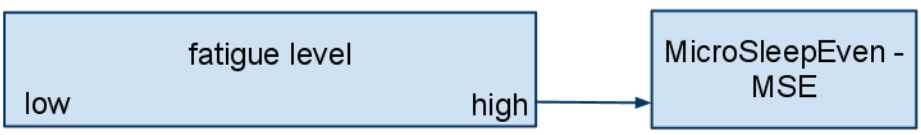


Figure 2.1 MSE detection

3. WORK

3.1 TESTED IMAGE PROCESSING ALGORITHMS

Various algorithms for head/eye/mouth localisation have been tested in this section. Different algorithms had been tested order to try if they are robust enough and fast enough for measuring the time of blink (3.3.6.1) under outdoor light conditions.

3.1.1 Mouth localisation

3.1.1.1 Algorithm description

The method [1] works with an image in the YCbCr (1) color model. The YCbCr color model separates luminance and chrominance components.

The main idea of the method is that lips are red-colored and then C_r is bigger than C_b. The lips color has a big response in C_r and small response $\frac{C_r}{C_b}$. Equations (2), (3) describes how to calculate Mouth Map. A μ is variable which is calculated from a local neighborhood Ω.

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -37.797 & -74.203 & 112.000 \\ 112.000 & -93.786 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

$$\mathrm{MouthMap} = \mathrm{C_r}^2(\mathrm{C_r}^2 \mu \frac{\mathrm{C_r}}{\mathrm{C_b}})^2 \quad (2)$$

$$\mu = 0.95 \frac{\sum_{(x,y)\in\Omega} \mathrm{C_r(x,y)}^2}{\sum_{(x,y)\in\Omega} \frac{\mathrm{C_r(x,y)}}{\mathrm{C_b(x,y)}}} \quad (3)$$

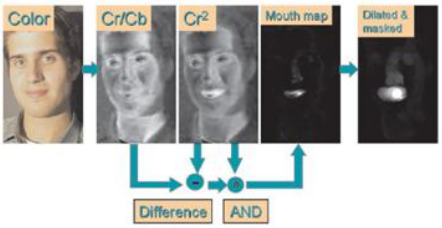


Figure 3.1 Calculation of MouthMap [1]

3.1.1.2 *Algorithm results*

This algorithm is easy to implement but is not robust enough for real applications without white balance (3.1.6). Another problem might be a lip painting. The results are shown in Figure 3.2.

Figure 3.2 Algorithm testing MouthMap

3.1.2 Symmetry detection

3.1.2.1 *Algorithm description*

A method [14] for the symmetry detection is based on comparing features points on an image. This method has been proposed to find the bi-linear symmetry and the rotational symmetry. The **S**cale-**I**nvariant **F**eature **T**ransform [35] has been used for detecting a features points on an image. The SIFT is invariant to the pose, rotation, scale and changes of illumination. The feature point is defined as a four dimensional vector of the coordinates, rotation and scale (x,y,φ,s). Any other features point extraction method (which generate four dimensional vector) for could be used.

The main idea of the the symmetry finding algorithm [14] is comparison of the position, scale, angle and descriptors of the feature points on the image. This method was tested in the BioID database with the accuracy of 95,1% [14].

Figure 3.3 Founded bilinear symmetry [14]

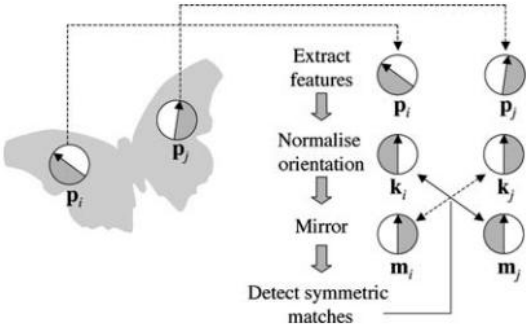


Figure 3.4 Features points comparsion

3.1.2.2 *Algorithm results*

The algorithm was tested in Matlab (Figure 3**.11**) with source code from authors of this method. This method has not been implemented in the final application because the method is not fast enought for measuring the time of blink (3.3.6.1).

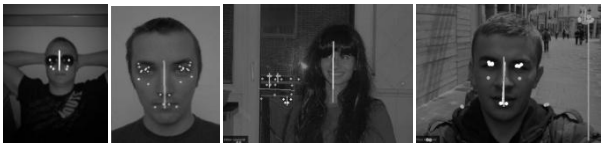


Figure 3.5 Founded bilinear symmetry

3.1.3 **Eyes reflection localisation**

- Algorithm description

This method is used to search eyes reflection on an image. Eyes reflection (a highly illuminated point on black background) localisation was used for many eyes localisations projects with an additional IR light [30]. Another projects [11] does't use additional light. The areas with these reflections are areas with high contrast in image. In frequential domain these are high contrast regions represented by high frequencies. I decided to detect these regions with method based on two dimensional **F**ast **F**ourier **T**ransformation.

FFT was calculated with Matlab function *fft2* and *ifft2*. After calling function *fft2* it is necessary to call fftshift which converts the image. This conversion moves

low frequencies to the center of the image and high frequencies towards the edge of the image. A binary image was used for bandpass filtrating. This is a simplification because the proper mask for fft2 filtrating should be calculated as 2D approximation of 1D filter (Butterworth, Chebyshev, Bessel). After function ifft2 I used abs value of the complex number instead of real part of complex number because the result was more acceptable.

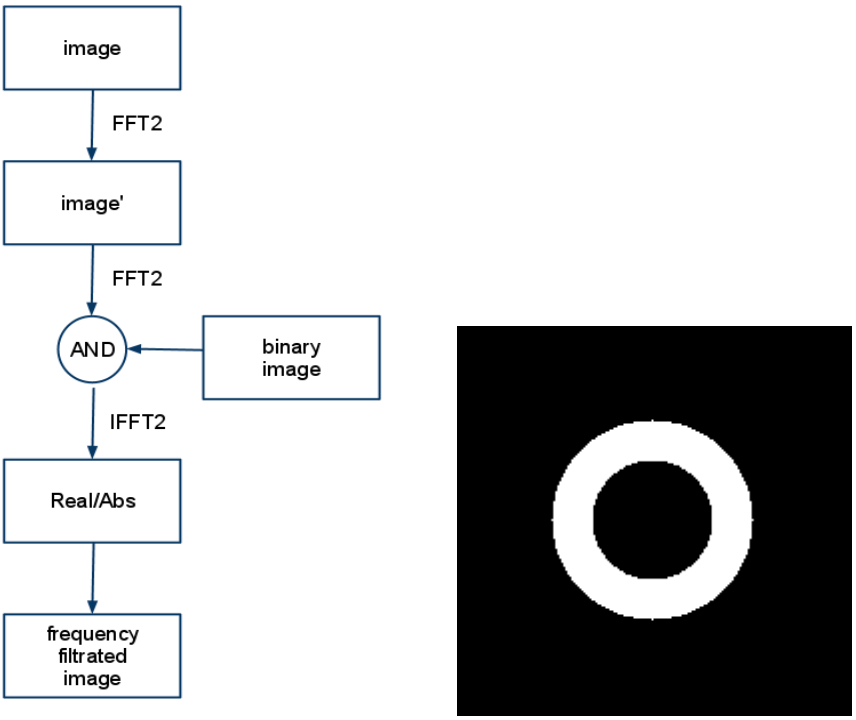


Figure 3.6 Band-pass filter a) algorithm b) fft template

- Algorithm results

The best eye reflections were obtained when the person looked straight into the camera and an additional light source was situated near the camera lens. In that case, the background around the eyes reflections is dark. Using an additional light source is a necessity for getting proper reflection in outside conditions. This method is unusable in a car because this project avoids using an additional light source. Convolution with a mask sized 5x5 and larger size is faster in frequential domain. Speed comparison between convolution in image domain and convolution by using

frequency domain can be observed in the Figure 3**.8**. Figure 3.7 shows more areas with high frequencies. This method is not reliable enough to find an eye in a complex image. On the other hand, using fft2 for finding the eye reflection gives better results than finding reflection with an edge detector – the response of an edge detector is worse than the fft2 filtration response Figure 3.7 b) and c).

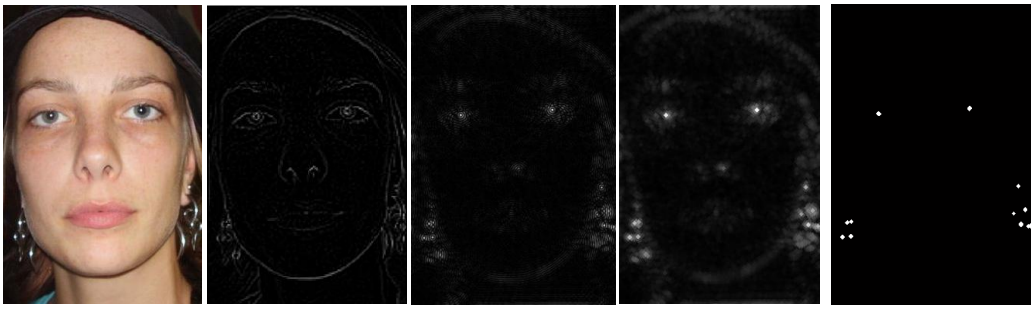


Figure 3.7 a) original imge, b) gaussian [3x3] edge detector, c) original image after fft2 filtration, d) dilatation with circle shape, e) thresholding

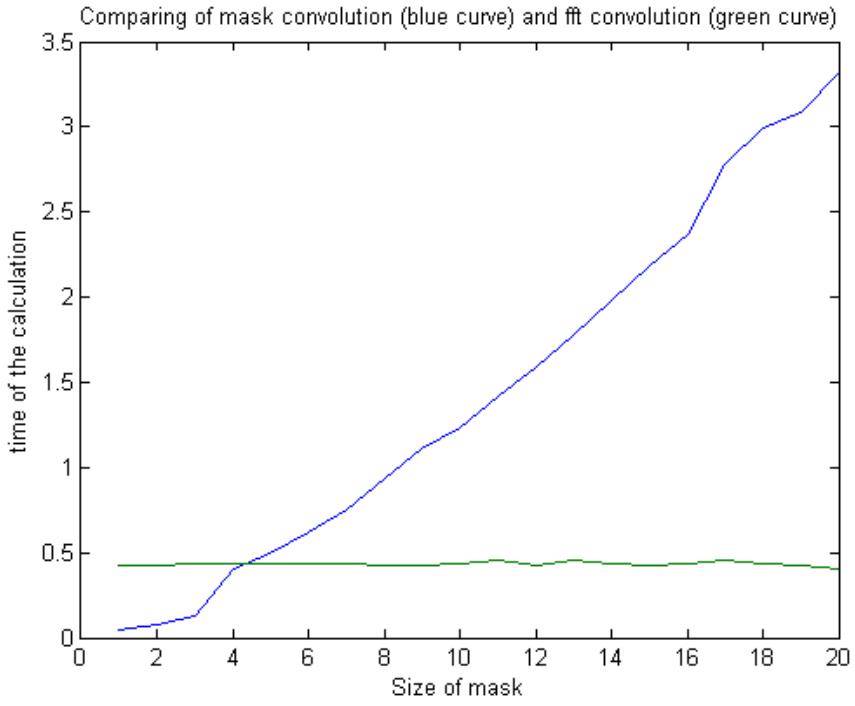


Figure 3.8 convolution speed comparsion

3.1.4 **Eye localisation**

Algorithm for eyes localisation is based on the three steps.

- EyeMap
- Head area segmentation
- Eye candidates reduction

3.1.4.1 ***EyeMap***

This method [1] works with an image in the YCbCr color model. The EyeMapC is calculated by the equation (4). The EyeMapC is a color transformation of Cb and Cr color channels. The EyeMapL is calculated by the equation (5). For calculation of EyeMapL morphological dilatation and erosion of Y channel is used. The morphological operations are defined for binary images but are generalised for their grayscale counterparts too. The morphological structure $g(x,y)$ with a circle shape was used for calculating EyeMapL. Then the EyeMapC is multiplied with the EyeMapL as is shown in Figure 3**.9**. After that EyeMap is normalised and tresholded (threshold is constant). This algorithm is based on the color information. For its correct working, the white balance (3.1.6) should be processed.

$$EyeMapC = \frac{1}{3}\left((C_b)^2 + (\overline{C_r})^2 + \left(\frac{C_b}{C_r}\right)\right) \quad (4)$$

$$EyeMapL = \frac{Y(x,y) \oplus g(x,y)}{Y(x,y) \ominus g(x,y)} \quad (5)$$

$\oplus$ Morphologic dilatation

$\ominus$ Morphologic erosion

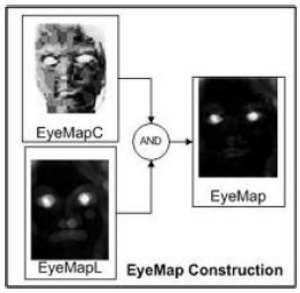


Figure 3.9 Calculation of EyeMap

3.1.4.2 *Head segmentation*

It is not possible to use this algorithm as stand-alone eye detector because a lot of noise is segmented too. A head detection is a possible way to improve accuracy of this algorithm – this is the reason why the EyeMap algorithm was tested together with the head detection algorithm (3.1.5.3).

3.1.4.3 *Eyes candidates reduction*

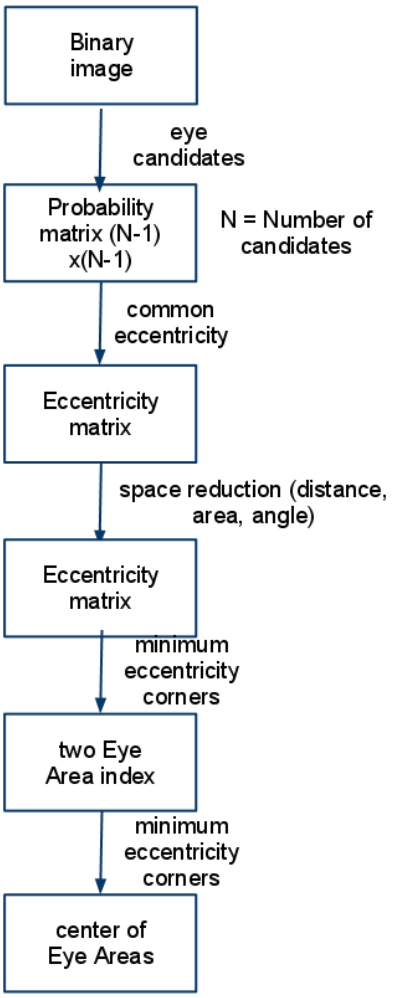


Figure 3.10 Eyes cendidates reduction

After the head detection and the eye detection, more eye candidates (Figure 3**.11** a)) could be segmented. The elimination of eye candidates is based on the similarity between the circle shape and the shape of eye candidates.

For all the candidates' eyes found in the head area, the eccentricity is calculated, with the eccentricity of an ideal circle being zero. Then, the eccentricity matrix is created. It is calculated as a product of two eye candidates' eccentricity. An element with a circle shape and the same radius as the eye iris radius was used for morphological operation dilatation and erosion.

The distance reduction is an important step because it eliminates a lot of the bad candidates. The minimal and the maximal distances of eyes have been calculated from the size of the segmented area.

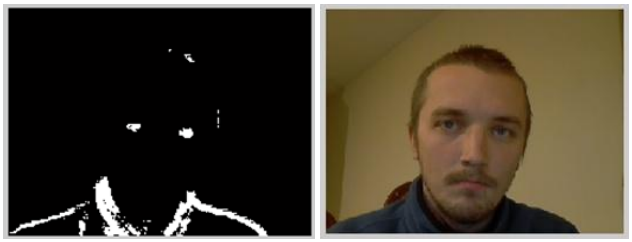


Figure 3.11 a) Eye map-thresholding, b) source image

Figure 3.12 Result of used algorithm

3.1.5 **Skin detection**

Two color-based methods are described in this section. The main advantages of this approach are low computing time, easy implementation and pose, rotation and scale independence. The main disadvantage of this approach is that the object with the same color as skin could be detected as skin. For its correct working, the white balance (3.1.6) should be processed.

3.1.5.1 *Takagi Sugeno fuzzy Interference system*

- Algorithm description

The fuzzy logic is not widely used for the image processing. The main goal of the experiment was to test the speed of the Matlab fuzzy toolbox. This method is listed [6] in the sources section. The algorithm is based on the Sugeno fuzzy model (a system without defuzzyfication). The method is based on eight IFTHEN rules and works with an image in the YCbCr color model 7.3. The inputs to the model are only Cb and Cr color channels.

*„**i.** If Cb is Light and Cr is Light, then the pixel = 0.*
***ii.** If Cb is Light and Cr is Medium, then the pixel = 0.*
***iii.** If Cb is Light and Cr is Dark, then the pixel = 0.*
***iv.** If Cb is Medium and Cr is Light, then the pixel = 0.*
***v.** If Cb is Medium and Cr is Medium, then the pixel = 1.*
***vi.** If Cb is Medium and Cr is Dark, then the pixel = 1.*
***vii.** If Cb is Dark and Cr is Light, then the pixel = 0.*
***viii.** If Cb is Dark and Cr is Medium, then the pixel = 1.*
***ix.** If Cb is Dark and Cr is Dark, then the pixel = 0.*

After that, we proceed to determining the degree of membership to appropriate fuzzy sets through membership functions. Once the inputs have been fuzzified, the final decision of the inference system is the average of the output ($_i z$) corresponding to the rule ($_i r$) weighted by the normalized degree pi of the rule.“ [6]

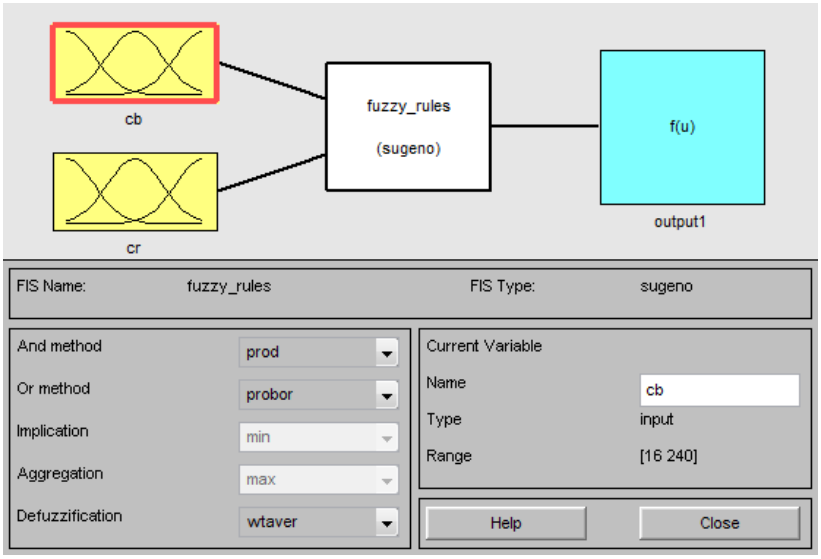


Figure 3.13 Sugeno model

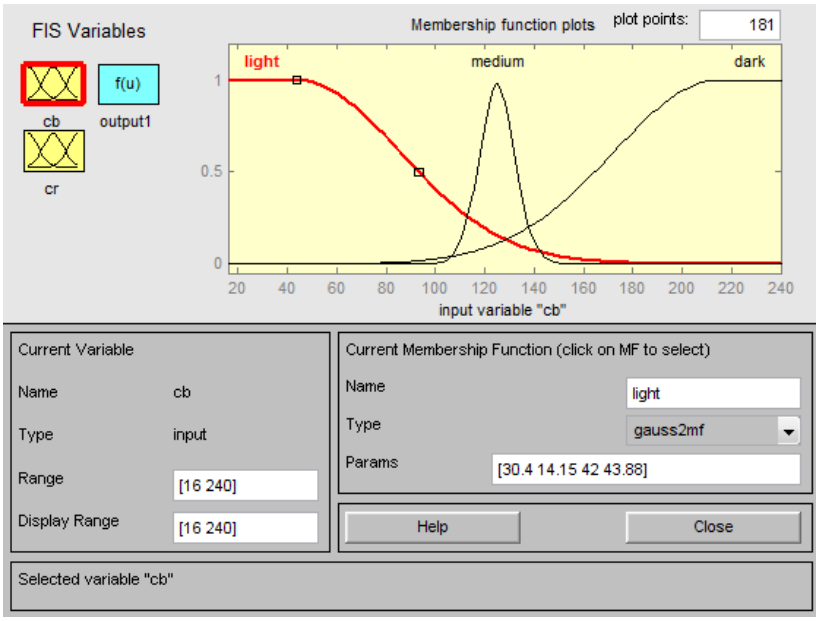


Figure 3.14 Fuzzification of Cb (same for Cr)

- Algorithm results

The implementation of this method is very simple. Its drawback is its long computing time which makes the Matlab fuzzy toolbox too slow for a real-time application.

Figure 3.15 Input image

Figure 3.16 Output image, time = 54.87s, size=302x202

3.1.5.2 *Skin color model*

3.1.5.2.1 Algorithm description

The Cb and Cr color channels of YCbCr color model (section 7.3) were used for describing the skin color. These color channels are independent on the intensity of light. The skin color of different people is placed in a very small area [35] (**Chyba! Nenalezen zdroj odkazů.**) in this color model. The YCbCr color model is widely used for skin detection [36].

Covariance matrix (6) is used for describing more dimensional random variables. On the diagonal there is a dispersion of each random variable. Covariance matrix is symmetric and positive-semidefinite too [13]. Matlab offers the function cov(). This covariance matrix was used to calculation of two-dimensional Gaussian function (7).

$$V_{i,j}^{m,n} = \langle (x_i \mu_i)^m \left(x_j \mu_j\right)^n \rangle \quad (6)$$

$$G(i,j) = exp(-0.5\, x' C^{-1} x) \quad (7)$$

C^{-1} $inverted$ covariance matrix

$x = [mean(cb) - i, mean(cr) - j]$

cb, cr – $color\ channel$

0.5 – $width\ of\ the\ base$

3.1.5.2.2 Algorithm implementation

The model was created from white skin samples. The **Figure 3.18** represents different skin samples. 70 different skin samples were used for creating a skin model. From these samples a two dimensional joint histogram (Figure 3.19) was made from the Cb and from the Cr color channel. From an image of different skin samples, a co-variation matrix that describes a more dimensional random variable was calculated. The matrix was used for calculating two dimensional Gaussian functions (Figure 3.20). The best skin model was obtained from the Skin samples 1 (**Figure 3.18** a)). The comparison of the different skin samples is in **Chyba! Nenalezen zdroj odkazů.**. Different skin samples are placed in a very small area. The intensity of illumination doesn't have effect.

3.17 Joint histogram for skin color Cb/Cr channel upper row, Approximation of joined histogram with gaussian distribution lover row; left lower corner [0 0], right upper corner [255 255], y axis Cb, x axis Cr

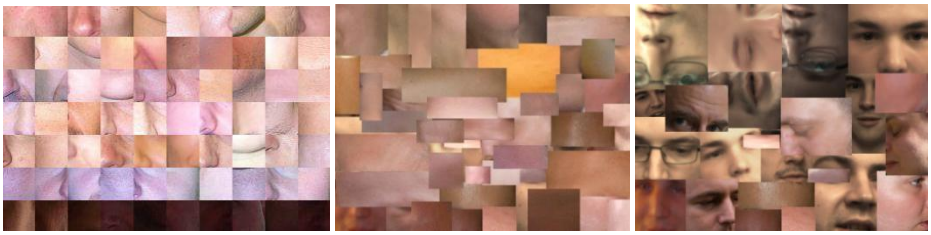


Figure 3.18 a) Skin samples 1 [15], b) Skin samples 2, c) Skin samples 3

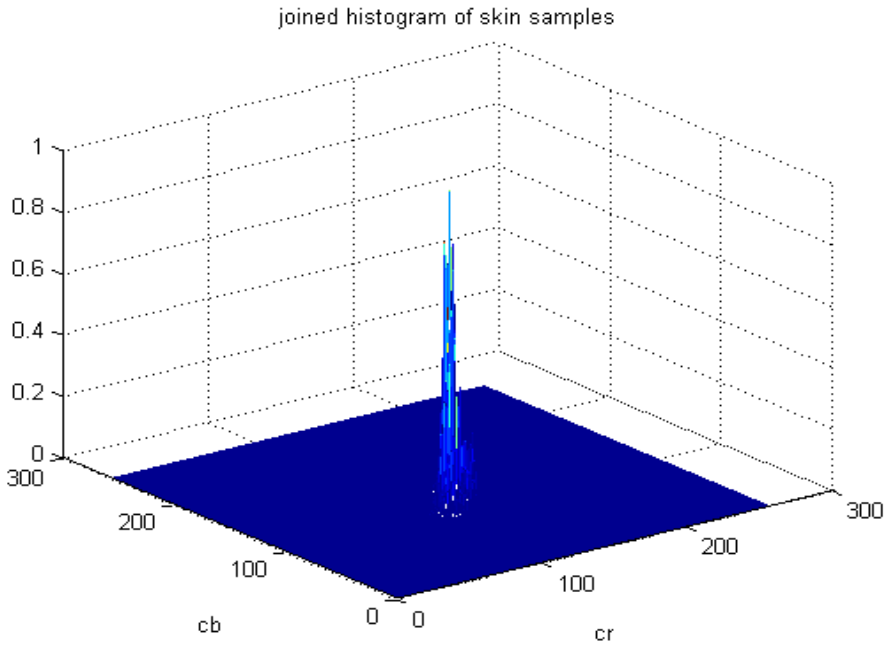


Figure 3.19 Joint histogram of Skin samples 1

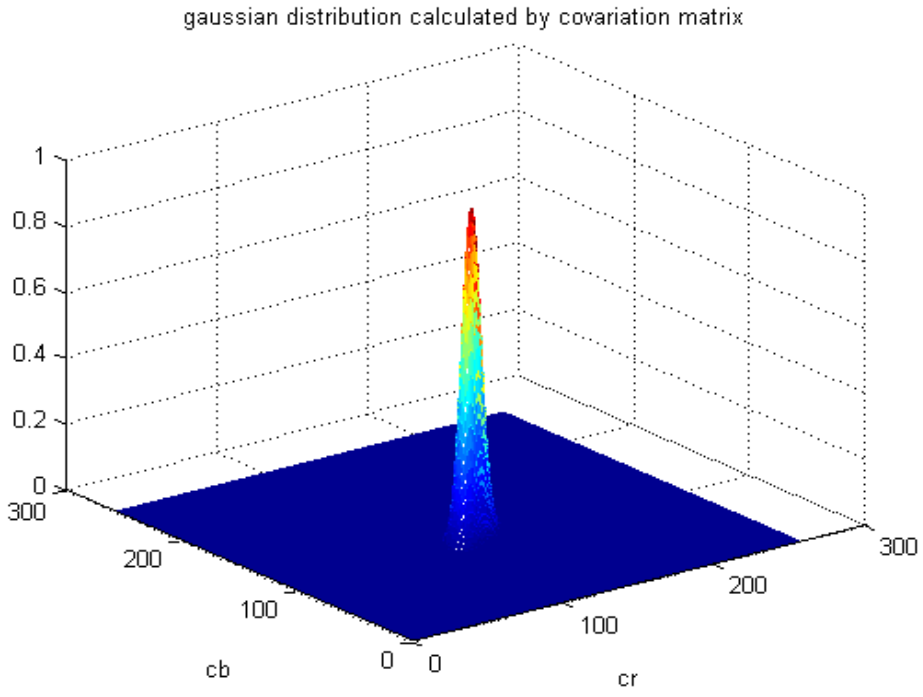


Figure 3.20 Gausian distributon of Skin samples 1

3.1.5.3 *Head detection with skin model*

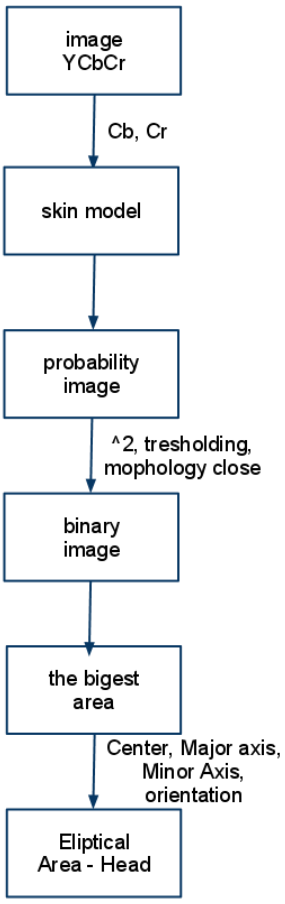


Figure 3.21 Head detection with skin model

The head detection with a skin model is a color based method. Both its advantages and disadvantages are discussed in this section.

The image is converted from RGB to YCbCr. Then, the response of a two-dimensional normalized skin model is calculated. The response is called probability map (probability that a pixel is a part of skin). Next, the probability map is blurred, thresholded and morphologically closed (Figure 3.22). Then, the object with the biggest surface is found. The biggest area is to be represented by the face, but in order to reduce noise it is more convenient to interpolate this area. The elliptical shape has been chosen for approximation. An approximated ellipse and original

shape has the same second order moment (center, mayor axis, minor axis, rotation to horizontal axis).

Figure 3.22 a) probability map, b) blur c) thresholding, d) morphology close

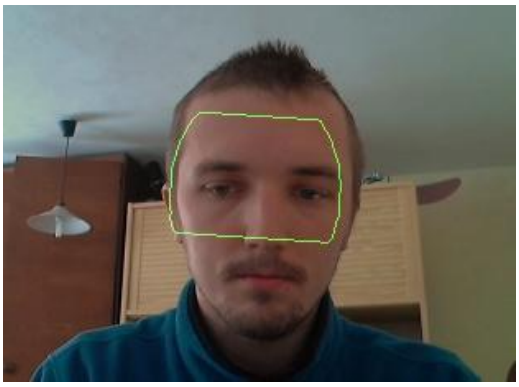


Figure 3.23 Detected head elliptic area with top and bottom cut off

3.1.6 **White balance**

The light compensation – white balance should be made for all color-based methods under changing light condition. The color of an object is influenced by the properties of the object's surface and properties of the light (mainly dominant wavelength). For a correct light compensation it is necessary to know the color of the light illuminating the object. Generally speaking, if we want to get a color of the light we have to know two things:

- color of the object under defined light condition
- color of the object under unknown light condition

As a consequence, there is a condition to be met: there must be an object with known color under defined light in the scene. The simplest way to obtain the color of light is to take a picture of the object with neutral color (white or gray).

We used a white paper placed in the center of the image. The color of white point has been extracted from the local area near the center of the image (Figure 3.**24**). Then light has been compensated by the equation (8).

A better way how to make a correct white balance is to send the color information about the white point to a camera. The camera will process the white balance faster and in the correct way (during image grabbing). It depends on the camera manufacturer but there are always non-linear transformations (sensor sensitivity, logarithmic conversion, gamma). The webcam (6.1) does not support any function for manual white-point.

Another algorithm for white balance in unknown image was presented in the source [1] . The white point is set as area of 5% pixels of image with the highest Y channel. This algorithm has not been tested.

$$IMG'(:,:,x) = \frac{IMG(:,:,x)}{C_x} \quad (8)$$

$$C_x \; color \; of \; white \; point$$

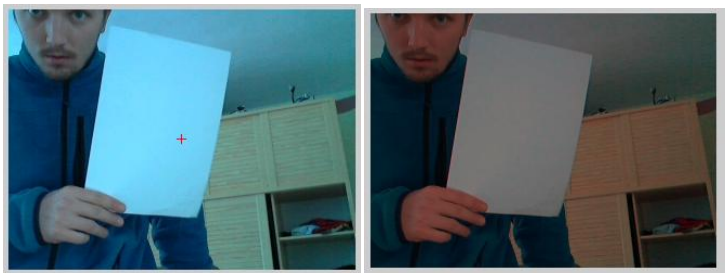


Figure 3.24 White balance with white point a) image before white balance red cross white point, b) image after white balance

3.2 PROPOSED ALGORITHM FOR HEAD DETECTION ON GRAY-SCALE IMAGES

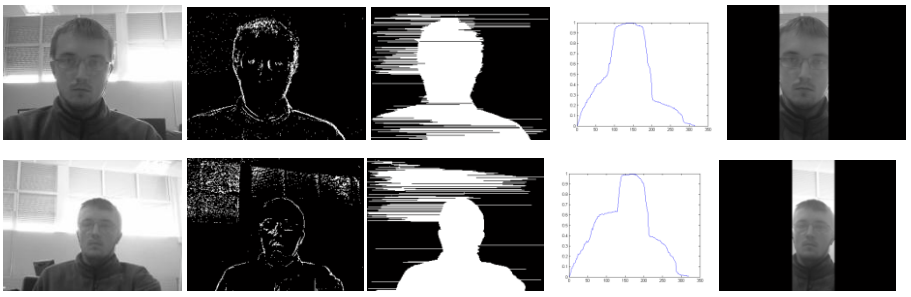
Using the algorithms (3.1) might be a problem under outdoor light conditions. None of them (3.1 algorithms) is robust enough and fast enough for measure time of blink (3.3.6.1) under outdoor light conditions. That's the reason why, a new algorithm for head detection in gray-scale images has been proposed.

The proposed algorithm for finding face on gray-scale images consists of three steps. The first step is a vertical head localisation, which is based on movement detection. The second step is a horizontal head localisation, which is based on seven face features. The last step is a template matching. The algorithms are described in detail in sections 3.2.1, 3.2.2 and 3.2.3. In section 3.2.4 is described implementation of proposed algorithm in OpenCV.

3.2.1 Vertical head localisation

The algorithm was proposed for video files. The main idea of the algorithm is very simple: it is based on a movement detection.

The limitation of this algorithm is that it localises only one person (one driver). On the other hand, small moving objects on the background are not a problem for this algorithm, nor is it necessary to move to make this algorithm work. The movements of our body cause bigger difference than a noise in differential image. The algorithm is paused in each cycle (when waiting for a movement). An acceptable result was obtained with a delay of 500 to 1000 ms. This delay would be handled with multi-thread access in a real aplication.



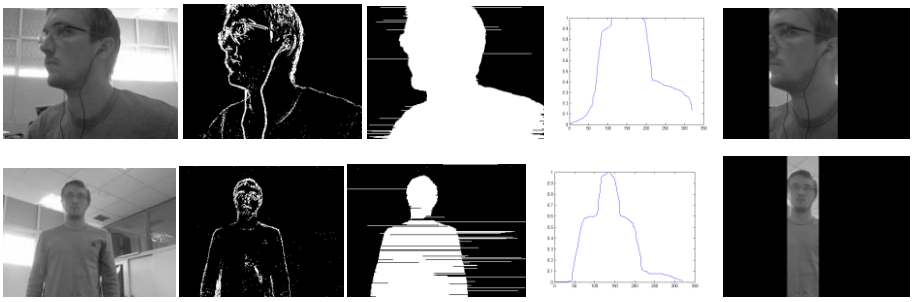


Figure 3.25 a) source image, b) differencial image, c) segmented image, d) sum of collums, e) head area

At first, we calculate a differential image. The differential image is the absolute value of substraction of two frames in a row. As shown on Figure 3**.25** b) there is noise too.

Second step is a dynamical thresholding. This step is very important. The calculation of the threshold value is shown on Figure 3**.26**. If threshold is very small there could be a lot of noise and segmented area could be the whole image. This bad result is eliminated by lowpass filter mentioned below Figure 3**.27**.

Thirdly, we process morphopigic dilation with circle mask and fill the horizontal line. Area between two points in one line is segmented.

Fourthly, we eliminate the areas which are nearly and fully overexposed or underexposed. These areas produce a lot of noise and if they are fully saturated they do not carry any information

Next, summing each row of image is processed. The curve we get is shown on Figure 3**.26** d). This curve is normalized and thresholded with constant threshold 0.8. Then the face position and width is calculated from that bool vector with the same length as the image's height.

Lastly, we apply a lowpass filter Figure 3**.27**, equation (9) for eliminating quick changes. We can do so because the speed of the head movement is limited and also position of head is quite stable during driving.

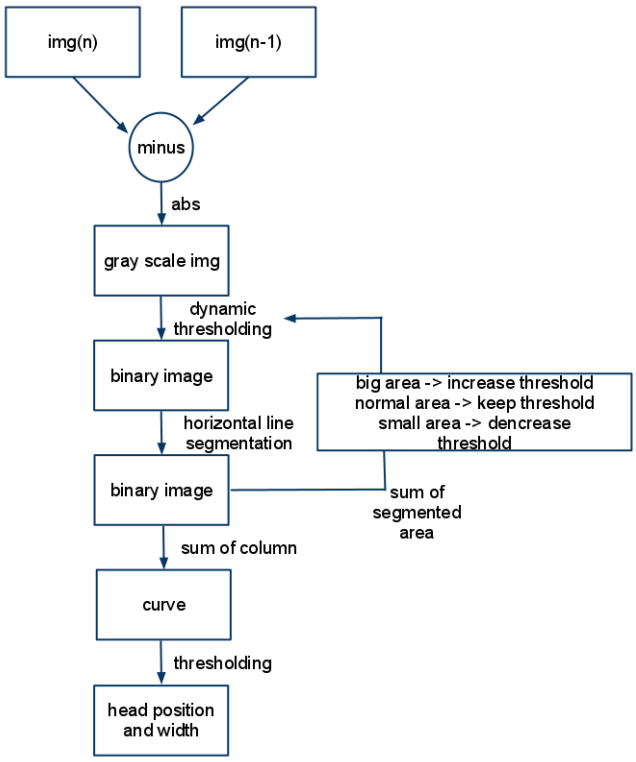


Figure 3.26 Head position and width calculation

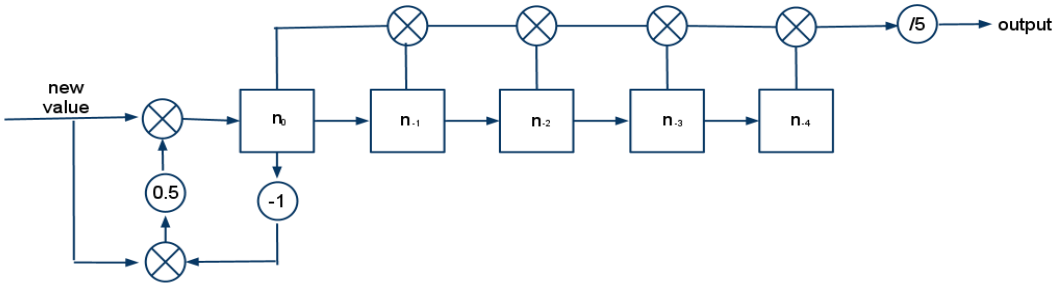


Figure 3.27 Lowpass filtr

$$output = \sum_{i=0}^{-4} n_i(9)$$

$$n_0 = new\ value\ +\ 0.5(new\ value - n_0)$$

3.2.2 **Horizontal head position**

The method for horizonal head localisation is based on the seven face/head features. Horizontal and Vertical features have been obtained both from the source. Other features have been obtained experimentally. The face features are movement,

head shape, haar features and variance of area. The Haar features and movement are calculated from an integral image.

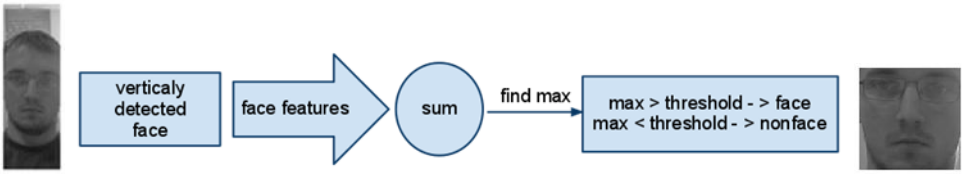


Figure 3.28 Horizontal head position

The integral image is a powerful tool when we want to calculate convolution over a big area. We only need three sum operations for sum area of any size.

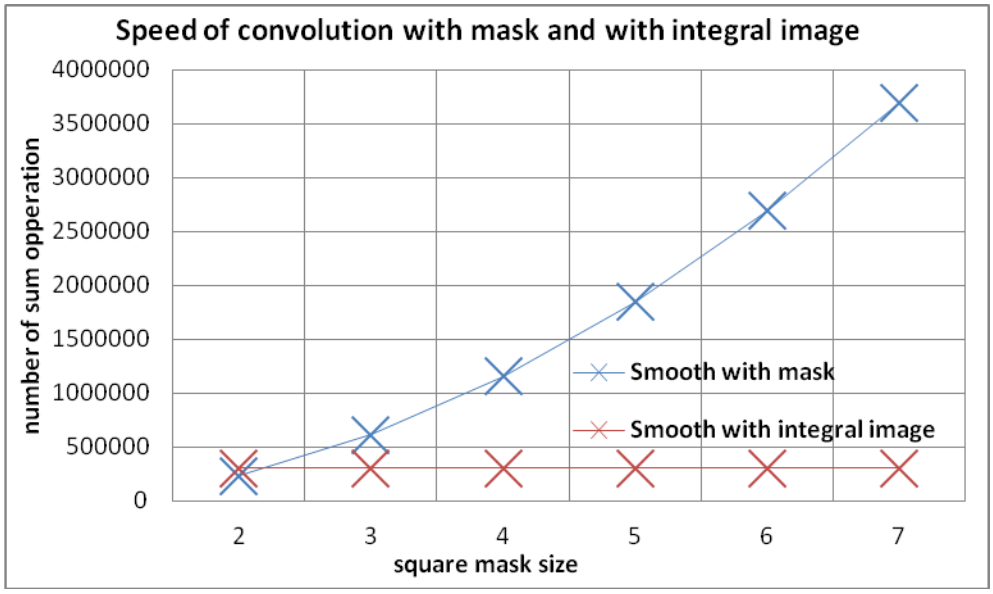


Figure 3.29 Speed of convolution and integral image

„The value of the integral image at point (x, y) is the sum of all the pixels above and to the left“

„The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. The sum within D can be computed as 4 + 1 − (2 + 3). “[21]

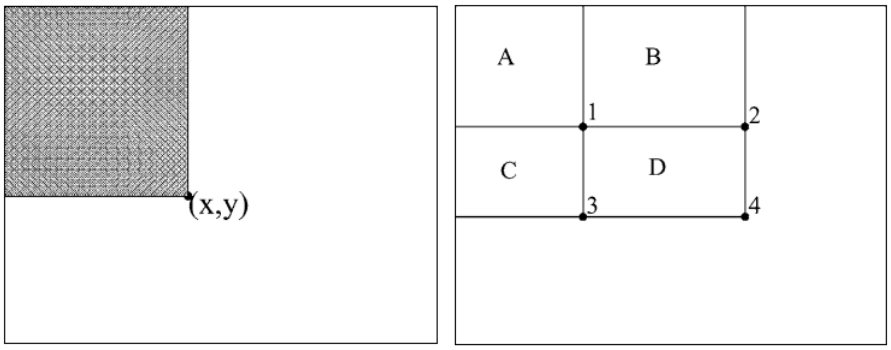


Figure 3.30 a) Calculation of integral image, b) calculation convolution from integral image [21]

3.2.2.1 ***Movement***

The integral image from the segmented image was calculated (Figure 3.31 a)), followed by a response to Haar wavelet (difference white area between black areas), shown on Figure 3.31 b). The response of black/white areas si normalised according to the size of black/white areas.

Figure 3.31 Feature Movement a) segmented image, b) haar feature

3.2.2.2 ***Head shape***

The response to Sobel edge operator from source image was calculated. Then the convolution was calculated using head shape template shown on Figure 3.32.

Figure 3.32 Head shape template

3.2.2.3 ***Variance of area***

The variance of square area was calculated and compared to variance of normal face. By means of experiment, we obtained a value of variance of normal face – 0.6.

3.2.2.4 *Haar features*

Four haar wavelets were calculated. The sizes of wavelets depend on head size (3.2.1). The calculation is very fast because both the vertical position and the head size are known and integral image (3.2.2.1) is used.

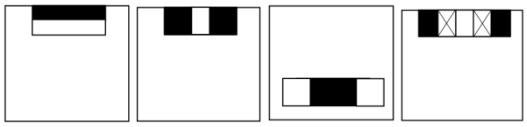


Figure 3.33 Haar features, shape/response a) horizontal [28], b) vertical [28], c) mouth, d) eye brow

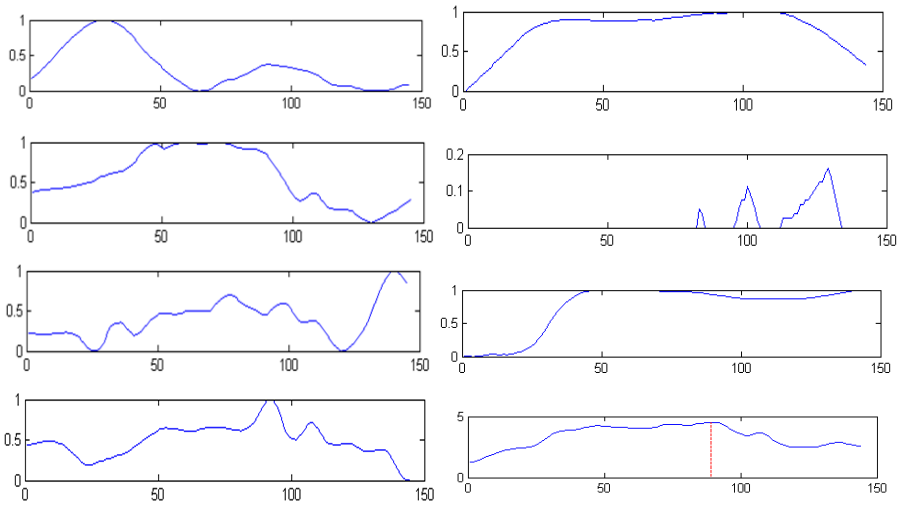


Figure 3.34 Typical feature response – left column haar features, right column – movement, shape, variance, final response with maximum

The performance of the algorithm might be increased by adding more face features. Clasificator with 7 face features is not reliable enough for face tracking.

On the other hand, this clasificator was not designed to detect a face in every frame. The algorithm may not detect face at every frame but the detected areas are very likely to be a face. The criterium of face/nonface is a value of the clasificator response (Figure 3.34). The the clasificator response is a sum of all the features (movement feature is multiplicated by two). The clasificator returns only faces with response value higher than seven.

3.2.3 **Template matching**

The template matching algorithm has been used for the face tracking. The template of face was obtained with the algorithm for finding faces in grayscale images (3.2.1). The face with the highest clasificator response is selected as face template. The cvMatchTemplate function from OpenCV library was used. The input to that function is a response of the Canny detector to template and response of Canny detector to a grabbed image. The correlation between a grabbed image and the template is computed by the equation (10) from reference [30].

$$R_{ccoeff\_normed} = \frac{R_{ccoeff}(x,y)}{Z(x,y)} \quad (10)$$

$$R_{ccoeff}(x,y) = \sum_{x'y'} [T'(x',y')I'(x+x',y+y')]^2$$

$$Z(x,y) = \sqrt{\sum_{x'y'} T(x',y')^2 \sum_{x'y'} I'(x+x',y+y')^2}$$

3.2.4 **Program in openCV**

The proposed algorithm was implemented in C++ using Visual Studio 2010 Professional and openCV v2.1. The application consists of two different approaches to face segmentation. Both algorithms are based on the algorithm for finding face in gray-scale image.

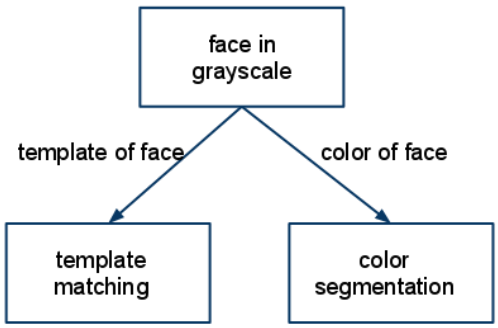


Figure 3.35 Grayscale face detection

3.2.4.1 *Finding the face in grayscale*

The algorithm was tested with a *usb* camera described in section 7.1. The camera was placed in front of the face on the notebook lid. The distance between the camera and the face was approximately 50 cm and the angle between horizontal line and lens axis was about 15°.

A person was watching video on the notebook for 15 minutes. During this time, 264 images of faces were obtained; the average grabbing rate was about 16 faces per minute. An average time of one iteration was about 40 ms (25fps).

Each cycle was paused for 100 ms. A delay of 100 ms was put to each algorithm cycle. This delay is important for vertical head (3.2.1) detection because it is based on movement detection.

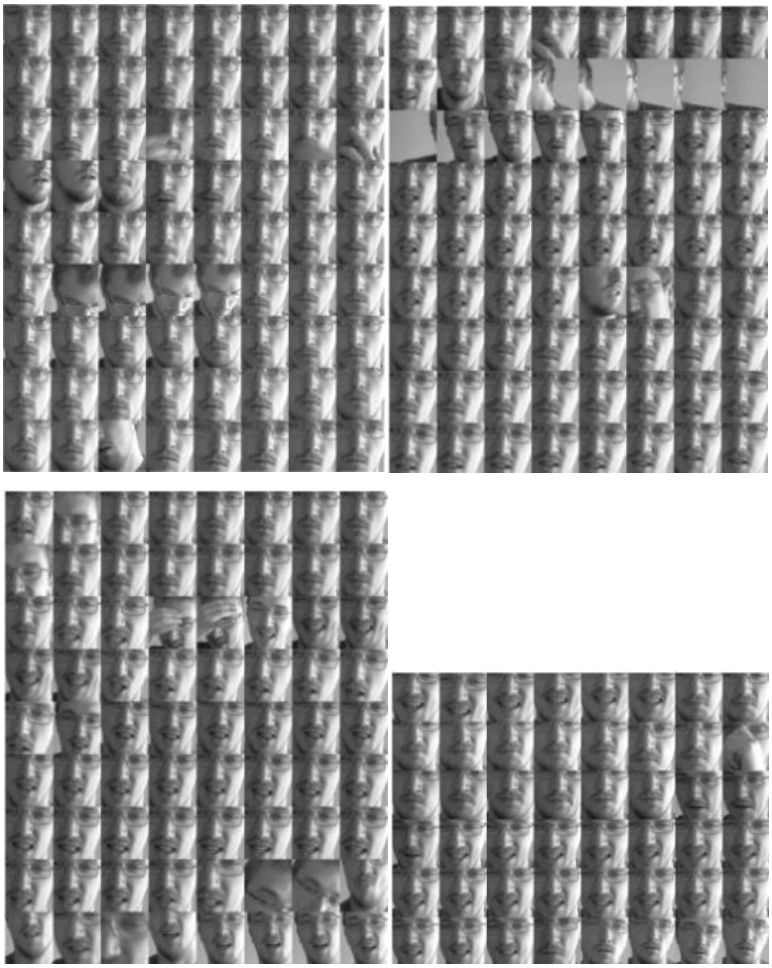


Figure 3.36 Grayscale face detection

3.2.4.2 *Face color segmentation*

The algorithm to face color segmentation is described in the section 3.6. The cvBlobsLib v8.3 library was used together with OpenCV. The cvBlobsLib is a library for regions labeling (an equivalent function to *matlab regionprobes*). There are some differences between the program in *matlab* and openCV.

The ellipse approximation of the head segmented area was not used. Instead, we used the morphological operation closing with rectangular shape. The size of the rectangle is changing due to the head width.

Another change is that instead of the Gaussian skin model dynamic *thresholding* was chosen. The threshold range depends on the face width.

The algorithm was tested only inside. The reason is that it is not possible to control all camera parameters in openCV. It is not possible to turn off auto-white balance, auto exposure and all other automatic functions which control color correction. All these automatic functions should be turned off. We have not included a different library in the project. The manufacturer of the camera does not provide any driver for image grabbing.

Another problematic issue with color segmentation is that a part of the head can be overexposed under outdoor light condition. The overexposed pixels do not carry any color information. This problem cannot be solved only by using YCbCr model because if all color channels (RGB) are overexposed, transformation to another color model is useless. This problem could be solved by controlling the camera exposition parameters. The exposition of the camera should be controlled according to the overexposed pixels in the face area. It is necessary to find a face in a gray-scale image and then set a proper exposition of the camera.

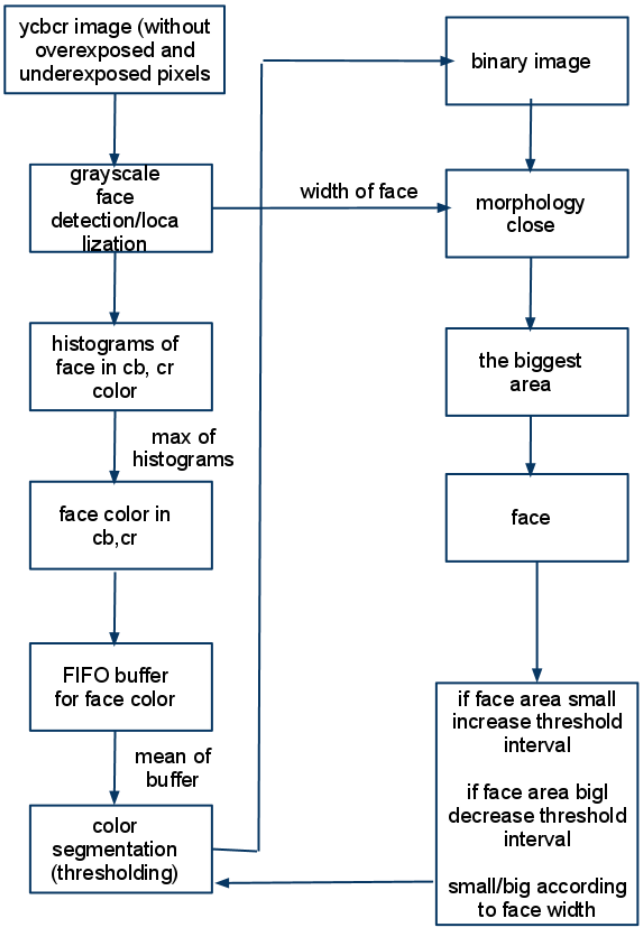


Figure 3.37 Face color segmentation

3.2.4.3 *Template matching*

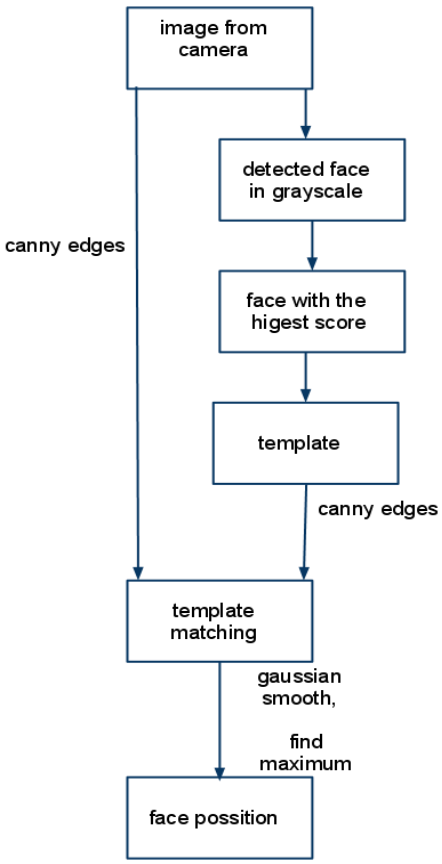


Figure 3.38 Template matching

This algorithm works with a frontal face image. The speed of the algorithm is about 10 fps in debug mode. The template matching is not rotationally invariant, but it works well if the angle between vertical axis and head axis is less than 15°. The scale in-variation has not been implemented, but it is not obligatory for car application. The method could be scale invariant because size of head is known

(3.2.1). This algorithm works well for gray-scale images and is invariant to strength of illumination. This algorithm is very robust, easy to implement and fast.

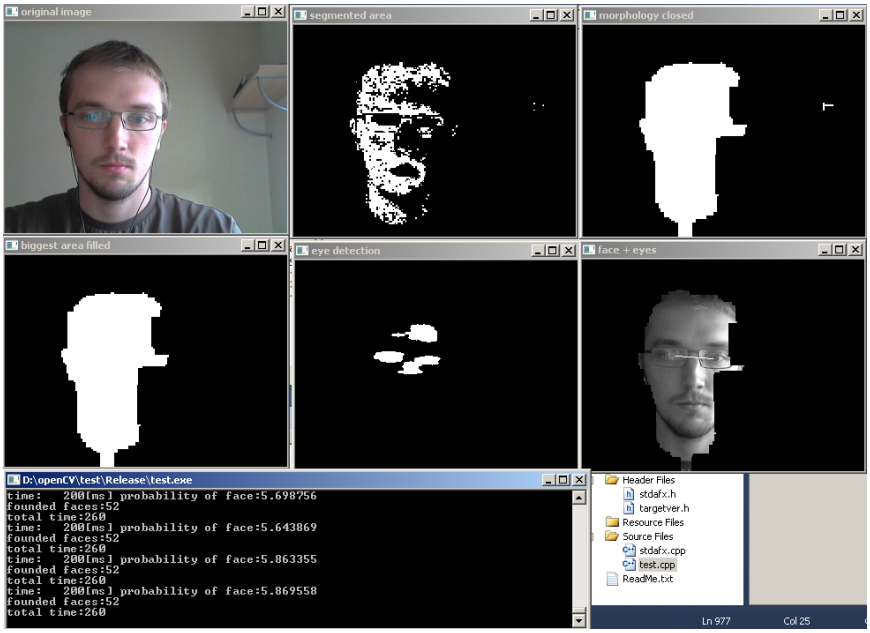


Figure 3.39 openCV aplication

3.3 EXPERIMENT – TIME OF BLINK, FREQUENCY OF BLINK

These facial fatigue features have been chosen because they are widely used [18] and their extraction is not very complicated in regard to the project‘s deadline. The reasons for measuring the duration of blink and frequency of blinking with a high frame rate camera (133 fps) was to provide evidence for a correlation between fatigue level and these facial fatigue features. Another reason for making this experiment was to determine the minimal possible fps for the camera to be used for measuring average eye open/closure time, time of closed eye during blinking and PERCLOS.

3.3.1 **Laboratory equipment**

As for the equipment, a personal computer with 2,8 i7 core processor and 4096 MB of RAM was used, along with a SICK Ranger C Camera and a DALSA X64-CL iPRO frame grabber capturing video recorded in laboratory. This device is

capable of a 133 frame per second rate in the resolution of 512x100 (gray-scale). We have used a SICK acquisition interface software for grabbing frames and OpenCV library v2.1 for saving video files to the harddisk. The video files were saved in uncompressed .avi format.

Figure 3.40 Experiment enviroment a) camera, b) frame grabber, c) power supply

3.3.2 The conditions of the experiment

Several experiments have been performed during the day and night. One person has been tested in different fatigue states, which were determined on a subjective basis. Relaxed 1 abd Relaxed 2 testing samples were recorded at midday so that the probability of being relaxed is high. Tired and Very tired testing samples were recorded after midnight when the probability of being tired is high. From all the video files parts with blink events were extracted using the VirtualDub-1.9.11 program.

3.3.3 Iris center detection

The Hough transformation and the morphological operations have been used for circle localisation (irises localisation). After that, a one-dimensional time-dependent signal representing the degree of eyes closing has been calculated below. All the video files were processed with Matlab. The speed of this algorithm was about 2,2 fps.

In this algorithm Hough transformation for irises detection has been used. Radius of iris was set manually so the output of Hough transformation is corners of the center (x,y). Finding the edges was realized by a well know sobel operator. I also

used eye morphology of the same kind as in EyeMap algorithm. The best result was obtained with a morphology element of circle shape with radius of iris. The found iris center is compared with the iris center found in the previous frame. I compared the distance between the two iris centers and the intensity of the circle area around the iris center. If the distance or change in intensity is greater than the threshold, the new iris center is abandoned and the old iris center is the correct result. The threshold for the distance is the iris radius.

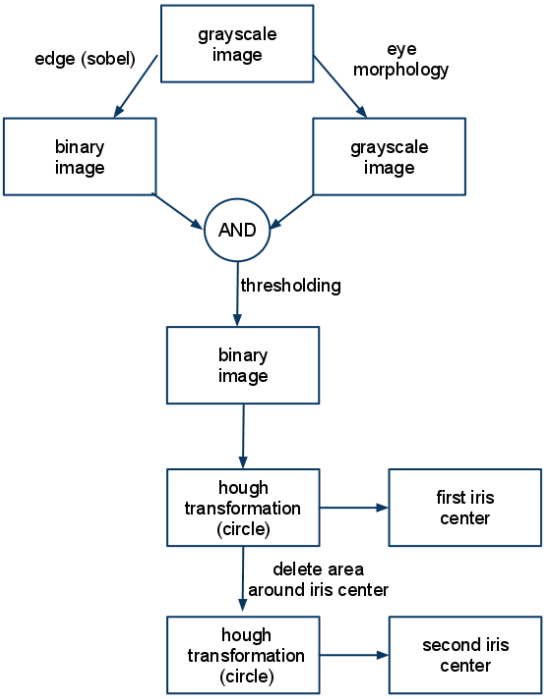


Figure 3.41 Iris center detection

Figure 3.42 Iris center detection

With this algorithm it is possible to localize the iris center of an opened and a closed eye. In order to do this it is necessary that both iris centers are localized correctly in the first frame of the video. Algorithm will fail if there is a fast movement of head during blink. In other cases algorithm works well.

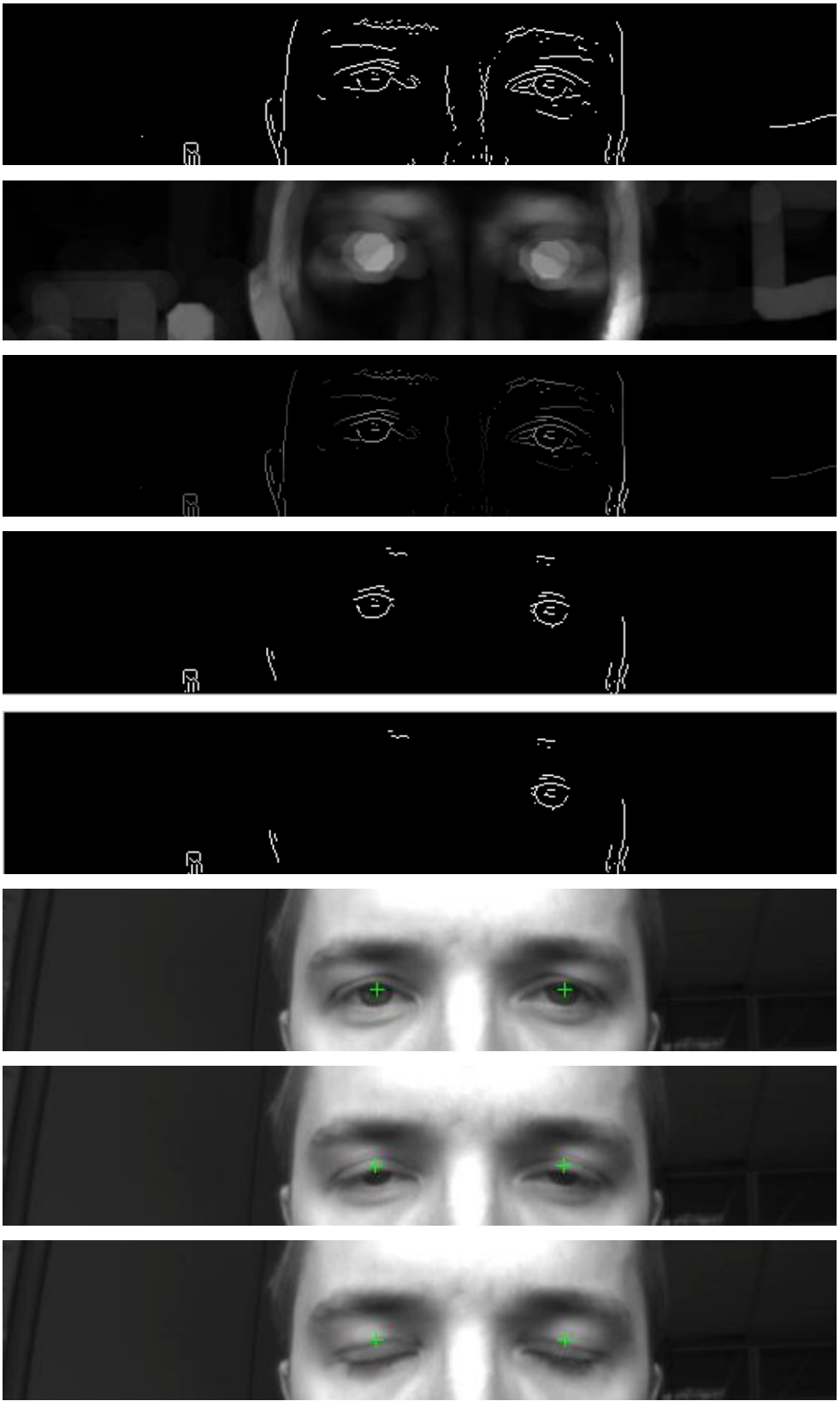


Figure 3.43 Iris center localisation

3.3.4 **Degree of opening**

The degree of the eyes opening was calculated as a sum of intensity in the circle area around the eye irises centers. The radius of the circle area was the same as was used in section 3.1.5. Result is average of the two eyes areas.

For the calculation of eye opening degree we have tried the method of template matching but the result obtained was not acceptable. Next method we tried to use was the thresholding of the eye area but it suffers from a big noise. All blink characteristics were normalized.

The one-dimensional time-dependent characteristics which represents the degree of eyes closing was interpolated with Matlab function for linear interpolation interp1. Interpolation was used in order to improve the accuracy of calculating time intervals (t_1, t_2, t_3, t_4). These times are described in section 6.5.2. These four times are used for calculating the time of eye closing (t_2-t_1), the time of closed eye during blinking (t_3-t_2) and the time of eye opening (t_4-t_3) and time of blinking (t_4-t_1).

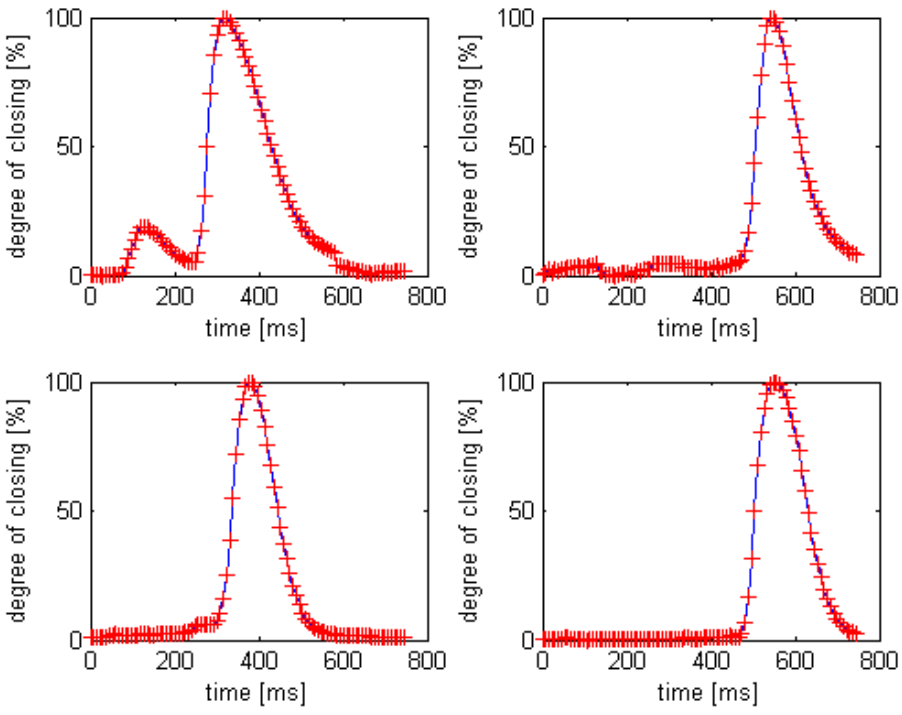


Figure 3.44 Diferent blinks state Relaxed 1

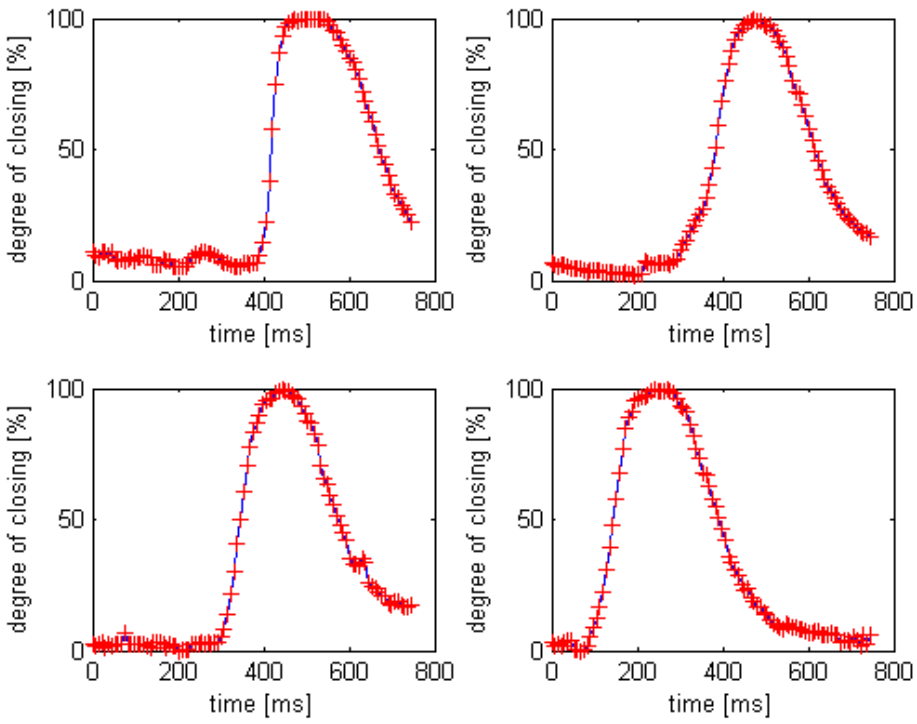


Figure 3.45 Diferent blinks state Very tired

3.3.5 **Frequency of blinking**

The frequency of blinking was also extracted. Another program for blink event detection has been coded for that purpose. It was necessary to create a fast algorithm capable of processing the thousands of frames. The program is based on movement detection and summing. This algorithm is very simple, fast and very robust. Its speed reaches about 50 fps.

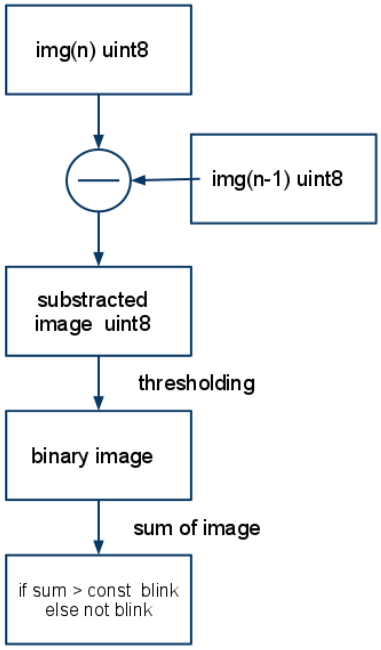


Figure 3.46 Frequency of blinking extraction

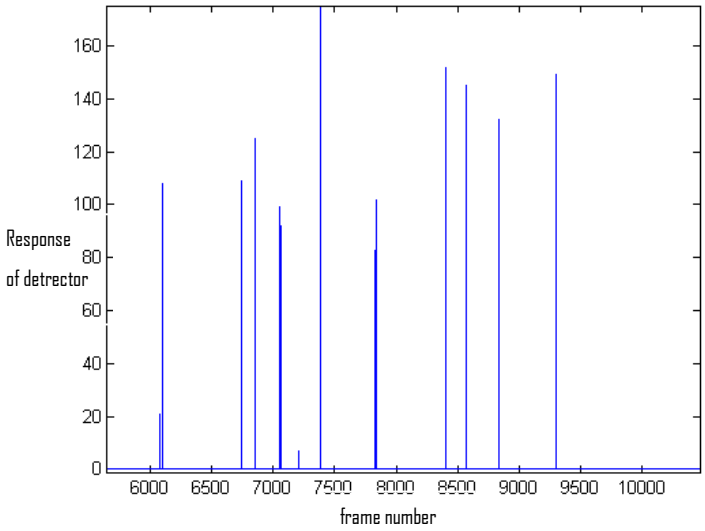


Figure 3.47 Typical response of blink detector

3.3.6 **The results of the experiments**

3.3.6.1 ***Time of blink***

The results of the experiments are shown as histograms in section 6.3. The histograms show the different times (t1, t2, t3, t4) of blinking for different fatigue levels.

Weighted averages of t1, t2, t3, t4 were calculated in order to remove noise by using the equation (11). The Gaussian function (12) was used for calculation of weights. It seems that all times (time of closing, eye closure duration, time of opening, time of blink) are growing with growing fatigue.

	Relaxed 1	Relaxed 2	Tired	Very tired
Time of closing [s]	0.0347	0.0219	0.0439	0.0612
Eye closure duration [s]	0.0792	0.0946	0.1081	0.1553
Time of opening [s]	0.0956	0.0693	0.1378	0.1436
Time of blink [s]	0.2157	0.1980	0.2969	0.3716
Number of blinks	23	36	17	36

Table 1 Weighted average of messured times

$$\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} (11)$$

$$f(x; \sigma, c) = \frac{-(x-c)^2}{2\sigma^2} (12)$$

	Time difference [s]	Fps
Time of closing	0,0393	50,9
Eye closure duration	0,0607	32,9
Time of opening	0,0743	26,9
Time of blink	0,1736	11,5

Table 2 Minimum Fps between Relaxed 2 and Very tired

The minimal fps (Table 2) which has to be used for measuring all blink times with regard to Shannon-Nyquist theorem was calculated. In Table 2 you can see the time difference between weighted average of state Relaxed 2 and state Very tired.

This experiment has proved that all times (t1, t2, t3, t4) of blinking are changing according to fatigue level.

3.3.6.2 ***Frequency of blinking***

The expected result of the frequency of blinking experiment was that the frequency of blinking would increase due to growing fatigue level. The results of the experiment were different. The time between a blink event was shorter in the state of Relaxed 1 (Table 1) and Relaxed 2 (Table 1) than in the states of Tired and Very tired. The histograms of the fatigue levels are shown in section 6.4. The main reason for these results might be that the video of relaxed states was grabbed during a sunny day. The eyes were drying faster than at night and started to burning. That is why the time interval between blink event was shorter during a day. The different results were expected. On the other hand, it seems that frequency of blink event is not influenced only by fatigue level.

4. CONCLUSION

4.1 TESTED IMAGE PROCESSING ALGORITHMS

The Matlab's fuzzy toolbox was tested for face color segmentation. The method is easy to implement. It consists of eight IFTHEN rules. The time of computation is very long, though; that makes using of this algorithm in real-time difficult. The algorithm is described in section 3.1.5.1. This algorithm was not implemented in the final application.

Another algorithm for face localization is based on symmetry detection. This algorithm compare features points obtained by the SIFT algorithm. The algorithm and the results are described in section 3.1.2. This algorithm was not implemented in the final application because an easier way for face segmentation was found.

Algorithms for mouth detection and for detection of eye reflections were also tested. However, these algorithms are not robust enough for real application. There are descriptions of this algorithms in sections 3.1.1 and 3.1.3.

Next, an algorithm for eye detection called EyeMap was also tested. This algorithm is based on color transformation from the RGB to the YCbCr model and a morphologic operation. It was tested under indoor light conditions with a common webcam (6.1). Detailed description of the algorithm is in section 3.1.3.

Because this algorithm is based on color information it is necesary to balance the white color correctly 3.1.6. This algorithm is not implemented in the final application because in openCV it is not possible to control the exposition of the camera.

4.2 PROPOSED ALGORITHM

We have the used C++ Microsoft Visual Studio 2010 Professional and openCV v 2.1 and cvBlobsLib v8.3 in the final application. Two approaches for face detection color segmentation and template matching were used. The extract of face color and face template was processed with the algorithm for face finding in grayscale image.

One approach is based on the color of human skin. The color of human skin is placed in a very small area in YCbCr color model. Advantages of color segmentation are invariation to scale and rotation, easy implementation and short computing time. Nevertheless, we encountered two issues. Firstly, incorrect white balance and secondly, overexposed pixels in the face area. Both problems could be solved by a possibility to control the camera exposition.

The other approach is based on template matching. Correlation is computed between two responses of Canny detector. The first one is a face template obtained by the algorithm for face finding in grayscale images. The second one is a grabbed image from camera. This algorithm works for grayscale images with the rate of 10 fps. Detailed description is in section 3.2.3.

The algorithm for finding the face in grayscale image works with a record of frontal face. The algorithm works at the rate of about 25 fps. This algorithm does not succeed in finding the face in every cycle but there is a high probability that the area found is the face. See the detailed description in section 3.2.

4.3 DRIVE FATIGUE

The main task of this work was extracting of driver fatigue characteristics. At the beginning of the project, a number of biological papers on correlation between driver's fatigue and visual characteristics were studied. The most important fatigue characteristics which can be extracted with camera is described in 6.5. It has been found that to detect a fatigue of different drivers we have to monitor more than one fatigue characteristics. We decided to measure blink duration.

Several experiments for measuring blink duration were done in laboratory conditions. The video was captured under different fatigue states (during the day and at night). The goal of the experiment was to prove a correlation between the time of blink and fatigue level and calculate the minimal possible fps which could be used for measuring blink duration. For this purpose we have used a grayscale camera SICK Ranger C and DALSA X64-CL iPRO frame grabber. The camera and the frame grabber were able to get images in the resolution of 512x100 with a rate of 133

fps. The video was divided into parts with blink events. A degree of opening characteristics from each video with blink was obtained. The algorithm is based on the Hough transformation and morphological operation. The correlation between the fatigue level and the time of blinking was proved. The time between blink events was measured but the results differed from our expectations. Description of both experiments and results are in section 3.3.6.

4.4 FINAL RESULT

The correlation between blink duration and the fatigue state has been proved with a high frame rate camera under laboratory conditions.

The algorithm for face localization and eye localization in color images was successfully tested indoors. Not all the problems with light compensation have been solved.

The algorithm for face finding in gray scale has been tested together with template matching with very good results. That algorithm is independent to color of illumination, intensity of illumination and scale. The algorithm is not entirely independent to head rotation. However, if the head rotation is smaller than 15°, the results are acceptable.

5. FUTURE WORK

The accuracy of the algorithm for face finding in grayscale could be improved by adding more Haar features. Haar face features make up one of the most powerful approaches for finding a face in grayscale images. The drawback is that the creation of a robust classificator based on haar features is not a trivial task.

It is necessary to propose an accurate low-cost eye center tracker which will be fast enough to register blink events.

Further research must be carried out on the correlation between driver's fatigue level and the responses of fatigue characteristics.

Also, it is important to consider whether the visual characteristics are reliable enough for preventing micro-sleep events.

The project of extraction of driver's fatigue characteristics could be simplified by using stereo-pair of cameras for obtaining 3D information and using additional IR light. Then, the location of the head and the eye could be solved more easily. Other characteristics like nodding, gaze, yawning, etc. could be obtained easily too. Another advantage of using two cameras is that we can increase fps.

6. APPENDIX

6.1 TESTING SAMPLES

The testing videos have been taken with the color camera placed in the vehicle interior. The camera has been mounted with rubber suction cup on the windscreen. The distance between the driver and the camera was approximately 50 cm. The testing videos have been taken in two different cars under different daylight.

- resolution 1024x768
- format .avi
- fps 10.
- auto-exposure disabled
- auto-white balance disabled
- compression disabled

The video of 30s of length is about 500MB large with these settings. Some of the samples have been overexposed in face area. The video was processed with VirtualDub-1.9.7 (rotation, changing resolution, file splitting).

Figure 6.1 Testing samples

For testing algorithms during implementation in Matlab and openCV, we used a usb web cam Logitech C 270 with the maximum resolution of 1280 x 720 with 25 fps.

All the image processing experiments were run on notebook with Intel core 2 duo T5470 and 2048MB RAM and Windows 7 x86.

6.2 DIVIDING OF FACE DETECTION METHODS [8]

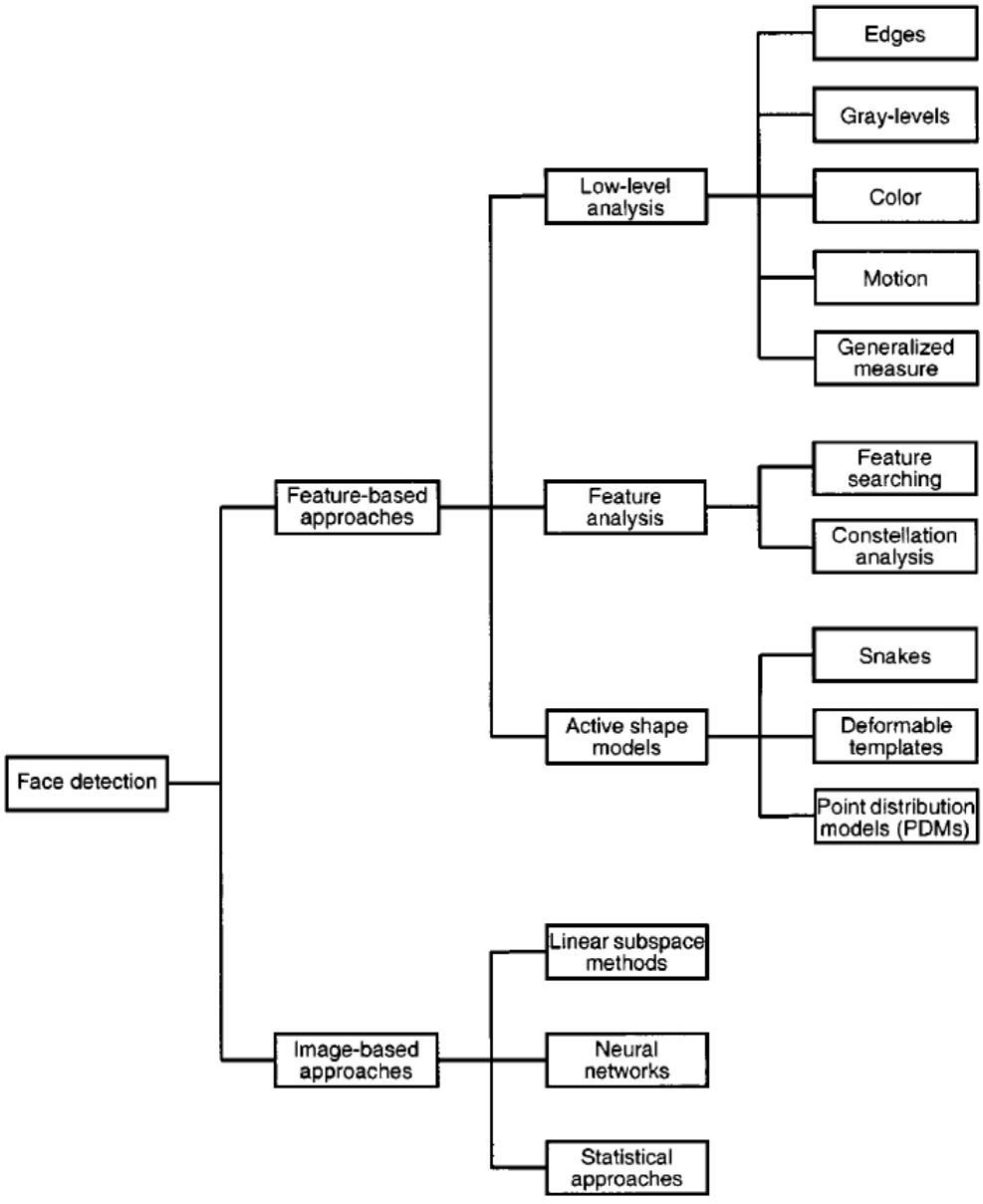


Figure 6.2 Face detection methods

6.3 HISTOGRAMS OF BLINKS CHARACTERISTIC

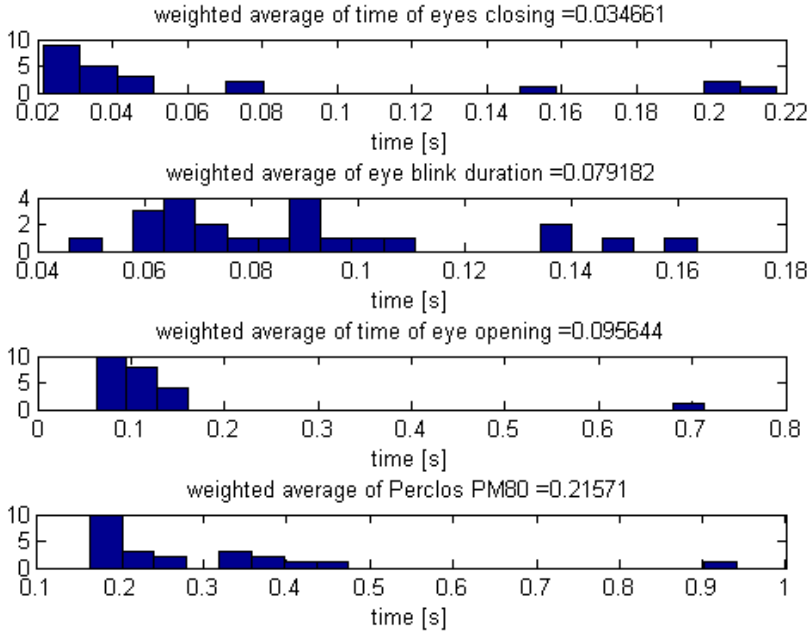


Figure 6.3 Relaxed state 1

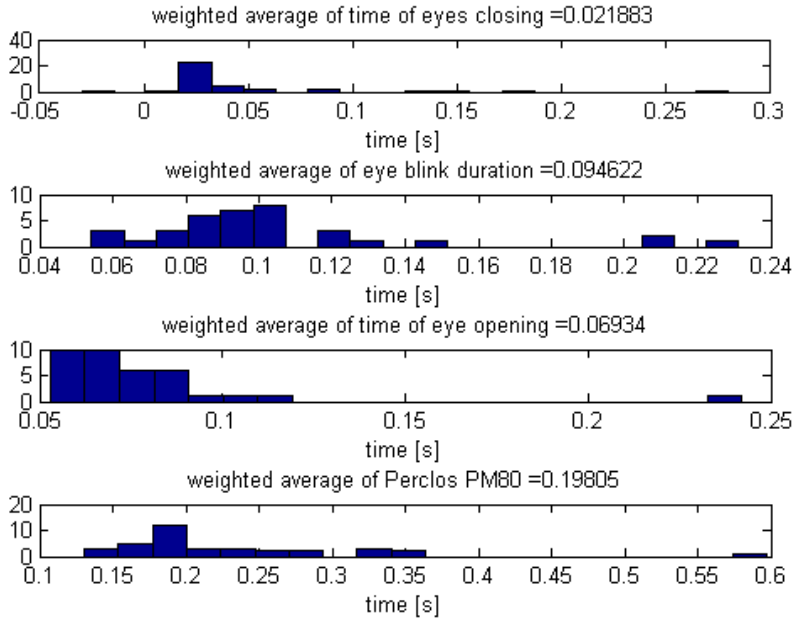


Figure 6.4 Relaxed state 2

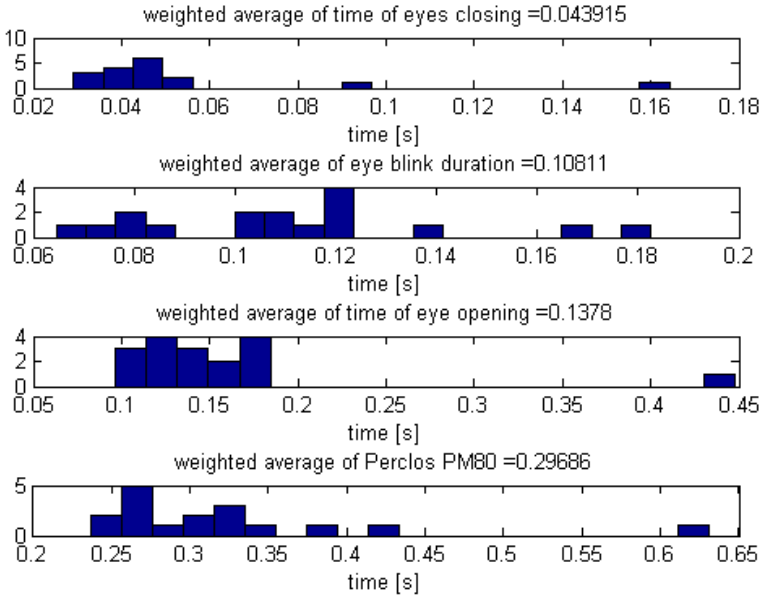


Figure 6.5 Tired state

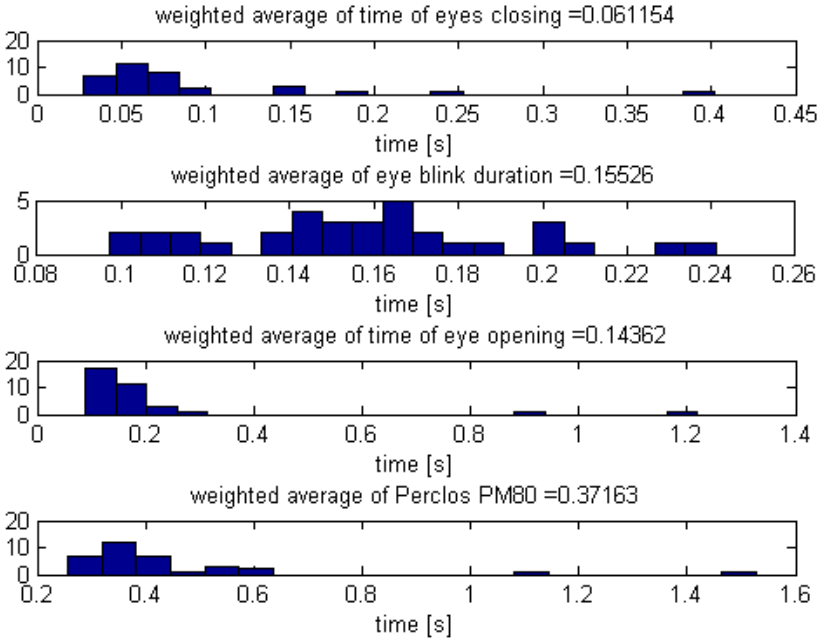
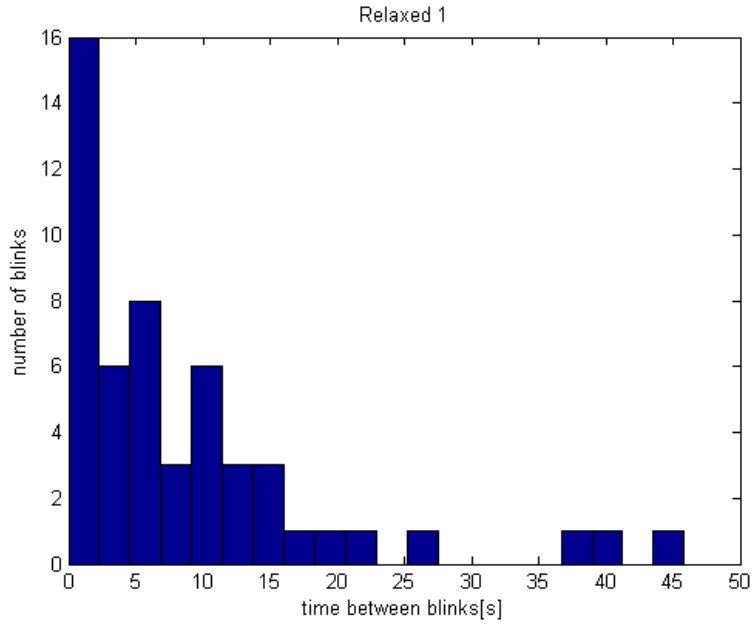
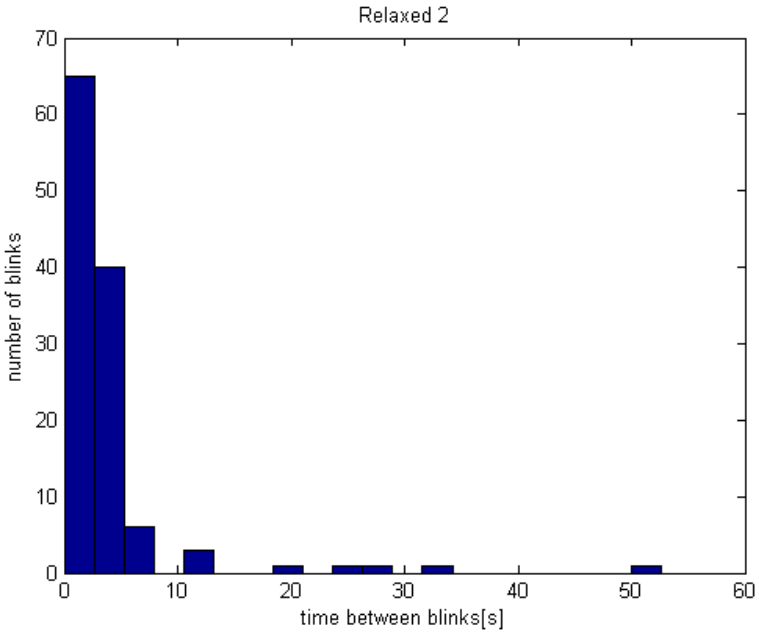
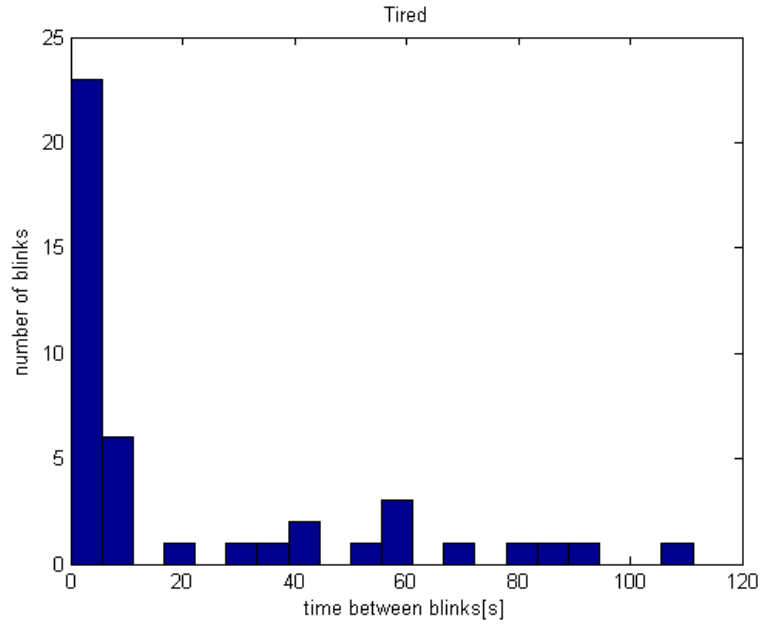


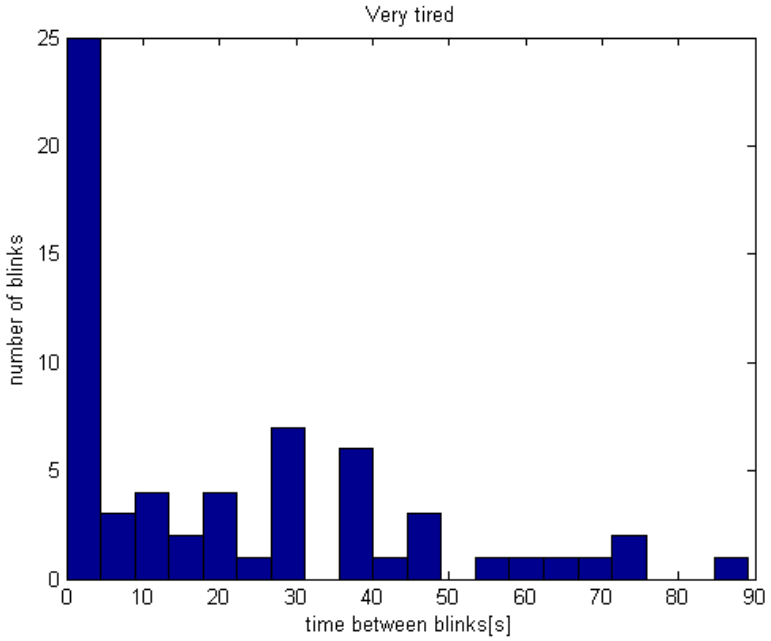
Figure 6.6 Very tired

6.4 HISTOGRAMS OF PERIOD BETWEEN BLINK EVENTS OF DIFFERENT FATIGUE STATE









6.5 FATIGUE CHARECTERISTIC

6.5.1 Average eye closure/opening speed

„Average eye closure/opening speed is computed as arithmetic average of all

eye closure speed over a fixed time period" [21]

6.5.2 **Perclos**

"PERCLOS is the abbreviation "Percent Eye Closure", is the time of the eyes closed refers to percentage of a specific time, and have P70, P80, EM three measurement methods. P80 is considered the best indicator of people's fatigue. Figure 46 shows the measurement principle of PERCLOS characteristic variables. Figure curve in the process of eyes closed and opened for one time in change with open degree, We can obtain the duration time of certain measuring extent of eyes closed or open according to the curve , and then calculated PERCLOS value.t1 is the time of the eye from fully open to closed 20% , For the eyes completely open to closed 80% is t2 , completely open to next open 20% is t3 ; the Completely open to the next open 80% is t4.By measuring the value of t1 to t4 ,We can calculate the value of PERCLOS."[21]

$$f = \frac{t_3 - t_2}{t_4 - t_1} * 100\ \%$$

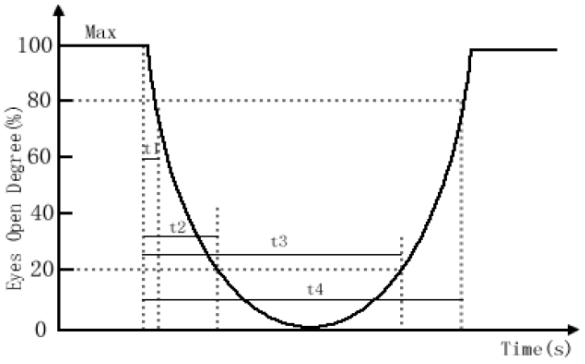


Figure 6.7 PERCLOS

6.5.3 **Eye blink duration**

"The eye blink duration is the time spent while upper and lower eye-lids are connected. ... the typical eye blink duration is less than 400ms on average and 75ms for minimum. For this reason we used T_{Drowsy} *=400ms and* $T_{Sleeping}$ *=800ms.*

Drowsiness Level	Description
Awake	Blink durations < T_{Drowsy}
Drowsy	Blink durations > T_{Drowsy} and Blink durations < $T_{Sleeping}$
Sleeping	Blink durations > $T_{Sleeping}$

6.5.4 **Blink frequency**

"Validation of Electrooculogram for fatigue detection using EEG and self-reported drowsiness. Blink behaviour changes with increasing fatigue, but large individual differences." [22]

„The majority of researchers agree that the rate of eye blink changes with the degree of mental workload, yet the question on how eye rate changes, still needs further investigation and clarification." [23]

6.5.5 **Pupilometry**

"Research has shown that there is a strong relationship between sleep deprivation, pupil size, and pupil stability. A well-rested individual can maintain constant pupil size in darkness for 15 minutes. As you become more sleep deprived, your pupil size will become less stable. It fluctuates (or oscillates) becoming subtly bigger and smaller rather than maintaining its size. Moreover, your pupils overall size will shrink, perhaps reflecting fatigue in the task of maintaining the larger size. Therefore, both pupil size and stability can objectively identify sleepiness and sleep deprivation. Pupillometry is not widely used, however, as it is mostly a research tool with the equipment not available much beyond this setting." [24]

6.5.6 **Iris excentricity**

"The degree of eye opening is characterized by the shape of pupil. It is observed that as eyes close, the pupils start getting occluded by the eyelids and their

shapes get more elliptical. So, we can use the ratio of pupil ellipse axes to characterize degree of eye opening." [18]

6.5.7 **Head position (nooding)**

"The head position sensor system MINDS (Micro-Nod Detection System) proposed by ASCI is conceptually designed to detect micro-sleep events occurring in association with head nodding by assessing the x, y, and z coordinates of the head through conductivity measurements. ... However, micro-sleeps also occur in the absence of obvious head nodding." [25]

"Head nodding monitors also exist however; this type of feedback may not alert the driver in ample time to prevent accidents and therefore, may not be useful for road safety application." [26]

6.5.8 **Eye gaze (blank stare)**

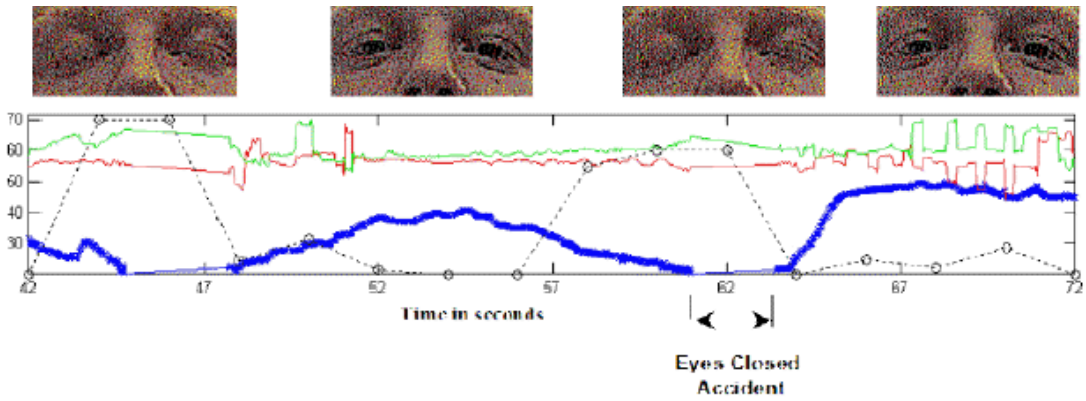
"Gaze has the potential to indicate a person's level of vigilance. A fatigued individual tends to have a narrow gaze."[38]

Eye gaze from 2D with IR light

6.5.9 **Detect Micro sleep even**

"Eye-gaze data appears to contain information about microsleep events several seconds before the real event takes place. Figure 6 illustrates a recording that includes a microsleep event. Approximately 10 seconds before the microsleep event occurs, the pupil diameter shows a slowly fluctuating pattern correlated to no change (blank stare) in the eye-gaze coordinates. Closer to the microsleep event the pupil diameter decreases and the eye-gaze coordinates are drifting until the eyes are closed. Especially interesting is the behavior of the eye after the second eye closure event, which was cut short by an accident. Immediately after the accident, a sharp increase in the pupil diameter occurred in connection with rapid oscillations in the

eye-gaze coordinates. This is the typical pattern for a person who is suddenly aroused by the accident and tries to re-orient himself."[25]



7. REFERENCES

[1] R.L. Hsu, M. Abdel-Mottaleb, A.K. Jain, "Face Detection in Color Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 696-706, May, 2002

[2] Jalal Aldin Nasiri, Sara Khanchi, Hamid Reza Pourreza, "Eye Detection Algorithm on Facial Color Images," Asia International Conference on Modelling & Simulation, pp. 344-349, 2008 Second Asia International Conference on Modelling & Simulation, 2008

[3] Paul T. Jackway and Mohamed Deriche. 1996. Scale-Space Properties of the Multiscale Morphological Dilation-Erosion. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 1 (January 1996), 38-51. DOI=10.1109/34.476009

[4] Kostka V. Extrakce příznaků z obrazové sekvence pro detekci únavy řidiče. Místo: VUT Brno. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, Rok vydání 2009. Ing.Karel Horák, Ph.D.

[5] Yang, G., & Huang, T. S. (1994). Human face detection in a complex background. *Pattern Recognition*, *27*(1), 53-63. Elsevier. Retrieved from

[6] Hmid, M.B.; Jemaa, Y.B.; , "FUZZY CLASSIFICATION, IMAGE SEGMENTATION AND SHAPE ANALYSIS FOR HUMAN FACE DETECTION,"*Signal Processing, 2006 8th International Conference on* , vol.4, no., 16-20 2006 doi: 10.1109/ICOSP.2006.345953

[7] ŽÁRA, Jiří. *Moderní počítačová grafika.* 2., přeprac. a rozš. vyd. Praha : Computer Press, 2004. 609 s., 16 s. barev. obr. příl ISBN 8025104540.

[8] E. Hjelmas and B. K. Low. Face detection: A survey. Computer Vision and Image Understanding, pages 236–274, September 2001.

[9] Ming-Hsuan Yang; Kriegman, D.J.; Ahuja, N.; , "Detecting faces in images: a survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.24, no.1, pp.34-58, Jan 2002
doi: 10.1109/34.982883

[10] PONZER, M. Detekce a rozpoznávání obličeje. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 84 s. Vedoucí diplomové práce Ing. Karel Horák, Ph.D.

[11] Rajpathak, T., Kumar, R., & Schwartz, E. (2009). Eye Detection Using Morphological and Color Image Processing. *Proceeding of Florida Conference on Recent Advances in Robotics* (pp. 1-6). Retrieved from

[12] Lamoš M, Cvičení č. 4 z předmětu MASO, Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010, E-Learnig

[13] Weisstein, Eric W. "Covariance Matrix." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/CovarianceMatrix.html

[14] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, Computer Vision – ECCV 2006, volume 3952, chapter 39, pages 508–521. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[15] Vlach J., metody a aplikace detekce mrkání očí s využitím číslicového zpracování obrazu, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008, vedoucí práce mgr. Pavel Rajmic, ph.d.

[16] Petyovský P., Přednáška kurzu MPOV - Klasifikátory, strojové učení,automatické třídění 1, , Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Brno, 2010

[17] Pavelka M., Vyhodnocení únavy řidiče na základě analýzy okulogramů s využitím programu matlab, Čvut, fakulta dopravní, laboratoř aplikované informatiky, Praha, 2005

[18] Liying Lang and Haoxiang Qi. 2008. The Study of Driver Fatigue Monitor Algorithm Combined PERCLOS and AECS. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering - Volume 01* (CSSE '08), Vol. 1. IEEE Computer Society, Washington, DC, USA, 349-352. DOI=10.1109/CSSE.2008.771

[19] Wu Qing, Sun BingXi, Xie Bin, and Zhao Junjie. 2010. A PERCLOS-Based Driver Fatigue Recognition Application for Smart Vehicle Space. In *Proceedings of the 2010 Third International Symposium on Information Processing* (ISIP '10). IEEE Computer Society, Washington, DC, USA, 437-441. DOI=10.1109/ISIP.2010.116

[20] Qiong Wang; Jingyu Yang; Mingwu Ren; Yujie Zheng; , "Driver Fatigue Detection: A Survey," *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on* , vol.2, no., pp.8587-8591, 0-0 0
doi: 10.1109/WCICA.2006.1713656

[21] Qiang Ji and Xiaojie Yang. 2002. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging* 8, 5 (October 2002), 357-377. DOI=10.1006/rtim.2002.0279 http://dx.doi.org/10.1006/rtim.2002.0279

[22] Svensson U., Blink behaviour based drowsiness detection, Publisher: Institution of Structural Engineers, ISSN: 0347-6049, Stockholm Sweden, 2004

[23] Husini, R.; Lenskiy, A.A.; Jong-Soo Lee; , "Distance learning and fatigue monitoring," *Strategic Technology (IFOST), 2010 International Forum on* , vol., no., pp.95-97, 13-15 Oct. 2010 doi: 10.1109/IFOST.2010.5667915

[24] B. Wilhelm, H. Wilhelm, H. Lüdtke, P. Streicher, and M. Adler. Pupillographic assessment of sleepiness in sleep-deprived healthy subjects. Sleep, 21(3):258–65, May 1998.

[25] Heitmann A, Guttkuhn R, Aguirre A, Trutschel U, Moore-Ede M. Technologies for the monitoring and prevention of driver fatigue. In: Proceedings of the First International Driving Symposium on Human Factors in Driver assessment, Training and Vehicle Design. Aspen, CO. 2001.

[26] Saroj K. L. Lal, Ashley Craig, A critical review of the psychophysiology of driver fatigue, Biological Psychology, Volume 55, Issue 3, 1 February 2001, Pages 173-194, ISSN 0301-0511, DOI: 10.1016/S0301-0511(00)00085-5.

[27] Danisman, T.; Bilasco, I.M.; Djeraba, C.; Ihaddadene, N.; , "Drowsy driver detection system using eye blink patterns," *Machine and Web Intelligence (ICMWI), 2010 International Conference on* , vol., no., pp.230-233, 3-5 Oct. 2010 doi: 10.1109/ICMWI.2010.5648121

[28] Paul Viola and Michael J. Jones. 2004. Robust Real-Time Face Detection. *Int. J. Comput. Vision*57, 2 (May 2004), 137-154. DOI=10.1023/B:VISI.0000013087.49260.fb

[29] Gary Bradski and Adrian Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Cambridge, MA, 2008.

[30] C. H. Morimoto, D. Koons, A. Amir, M. Flickner, Pupil detection and tracking using multiple light sources, Image and Vision Computing, Volume 18, Issue 4, 1 March 2000, Pages 331-335, ISSN 0262-8856, DOI: 10.1016/S0262-8856(99)00053-0.

[31] Winslow L., Truck Technologies of the Future, Online Think Tank, 2007

[32] Jan T., Don't sleep and drive – VW's fatigue detection technology

[33] Williamson, A. (2005). Review of on-road driver fatigue monitoring devices.*New South Wales Australia NSW Injury Risk*.

[34] Meng, Y., & Tiddeman, B. (2008). Implementing the Scale Invariant Feature Transform (SIFT) Method. *Citeseer*. Citeseer. Retrieved from doi=10.1.1.102.180&rep=rep1&type=pdf

[35] Amine, A., Ghouzali, S., & Rziza, M. (2002). Face Detection in Still Color Images Using Skin Color Information. *Third International Conference on Emerging Trends in Engineering and Technology*, *24*, 696-706.

[36] Jorge Alberto Marcial Basilio, Gualberto Aguilar Torres, L. Karina Toscano Explicit image detection using YCbCr space color model as skin detection. In *Proceedings of the 2011 American conference on applied mathematics and the 5th WSEAS international conference on Computer engineering and applications*, Alexander Zemliak and Nikos Mastorakis (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 123-128 [37]

[37] Mita, T.; Kaneko, T.; Hori, O.; , "Joint Haar-like features for face detection," *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* , vol.2, no., pp.1619-1626 Vol. 2, 17-21 Oct. 2005 doi: 10.1109/ICCV.2005.129

[38] Zhiwei Zhu; Qiang Ji; , "Real time and non-intrusive driver fatigue monitoring," *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on* , vol., no., pp. 657- 662, 3-6 Oct. 2004 doi: 10.1109/ITSC.2004.1398979