LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

Faculty of Technology

Master's Degree Programme in Technomathematics and Technical Physics

Author of the thesis Galyamichev P.

“Traffic signs detection method”

Examiners: Heikki Haario (Ph. D.)

Supervisor(s): Tuomo Kauranne (Ph. D.)

2010

Table of contents

Introduction

Today a car is an essential part of our life. In the XXI century a car is waiting for changes because it must be not only a conveyance but an assistant on the road. In 2007 there were more than 230,000 road accidents in Russian Federation. During 7 months of 2009 there were more than 100,000 accidents. A large number of accidents occurred due to carelessness of drivers and traffic violations.

According to modern requirements for vehicle safety there must be noted that effective driving is often more dependent on computer systems that these vehicles are equipped with rather than the driver. Therefore the main measure to reduce accidents on the roads is to be done in the development of systems that provide control of the vehicle while moving.

To solve this problem, we plan to develop an automated system that would allow to detect traffic signs along the way, in advance informing the driver about changes in driving conditions without detracting from driving. Thus the driver will not be distracted from driving when special vigilance is required [1].

The aim of this Master Thesis is to develop a method for traffic signs detection for vehicles that reduces the number of accidents while driving. This method will be developed as an automated software-hardware solution that will be supplied with the vehicle.

In order to automate the process of traffic signs detection and warning driver about his position in traffic flow it`s necessary to solve a few problems [1]:

- analyze the systems existing at the moment;
- justify the choice of the developed method and software;
- formalize the calculations;

- justify the development of all types of software;
- build an infological model;
- implement the selected option of the project;
- make an experimental study.

The main advantage of the automated system of traffic signs detection is a reduction of the risk imposed on the driver of the vehicle while driving, as well as increasing the information content.

This method is expected to be implemented in vehicles as an integrated system. The user of this method will be the driver who will have to get visual and audible warning when the traffic sign is on the path of the vehicle [1].

Automating traffic sign detection allows us to reduce the amount of time necessary for appreciating the state of movement and actions of the driver for any manipulation.

Creating your own system, will allow to take into account all the features of currently existing developments, and to minimize the redundancy of procedures and increase the efficiency of the system developed.

1. Analysis of existing systems and methods for traffic signs detection

1.1 The problem of allocation of traffic signs

There are a huge number of systems and methods to detect and recognize the traffic signs on the road, for warning the driver. They all have some common points in their structure, but also significant differences that distinguish these methods among other developments.

The general structure of such methods can be summarized as follows:

- receiving video stream from an external device;
- preprocessing stage:
    - allocation of the image, which supposedly contains a traffic sign;
    - allocation of the image, which contains a traffic sign;
    - extraction of the geometric shape, which is a traffic sign.
- the traffic sign recognition stage;
- output of the audio and visual information on the recognized traffic sign.

Let us consider all the stages of these systems in detail.

First, it is necessary to select the hardware equipment to solve this problem. Typically, these methods use a camera with a refresh rate of about 20 frames per second. In practice, this frequency is sufficient even for the real time systems.

The second stage is based on color processing, or object detection method based on rapid color changes. Color processing takes much time and requires many steps of image processing, which affects both the time of image processing, and hardware downloading. In this case there is a high probability of error when working in low visibility conditions or at a noise when receiving

a picture. Image processing technology for rapidly changing targets is mostly used when working with black and white images. It is more efficient and fast also in processing of color images, but it also has some limitations. This stage is based on a set of standard filters and licensed methods, such as: the Gaussian filter, the method of selection of geometric shapes, brightness and contrast, methods of segmentation and prediction.

The third stage of these methods does not have a wide range of possibilities. As a rule, modern systems use pattern recognition technology with construction and training a neural network. The system receives the processed image containing the traffic sign at its input area, and gives a possible identifier as like a traffic sign at the output. Using neural networks is very effective for object recognition, but inquiry a high cost in hardware resources.

The procedure of audio and visual information output, has no common features with the other part of the system, and therefore its description is not required.

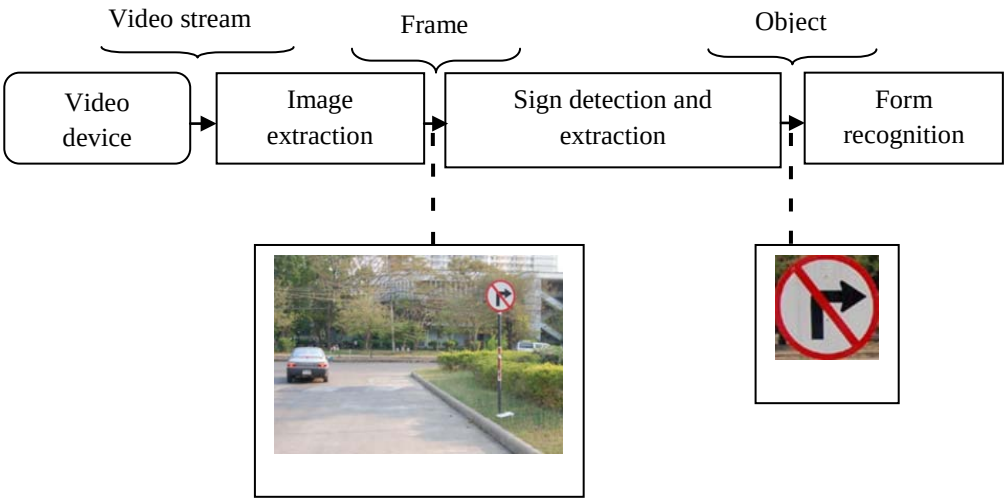
1.2 Description of existing methods of the traffic signs detection

1.2.1 Detection method using a morphological analysis

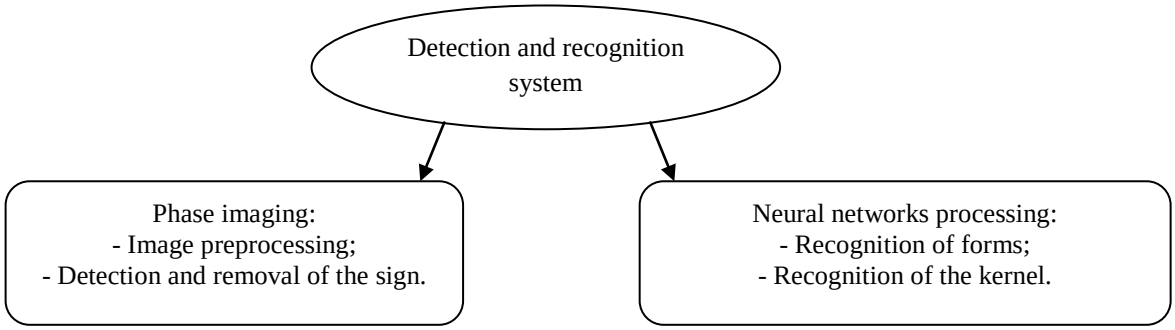
One of the most progressive and fastest growing techniques is the detection using Gaussian filter and morphological analysis. This method provides the procedure for detection and recognition in real time, which undoubtedly is an advantage because it allows the user to obtain relevant information. Graphically, the basic configuration can be represented as follows (pic. 1) [2].

Detection and recognition of traffic sign stage on the obtained image is a separate module, divided into two streams: the first stream is actually the stage

of the image processing, the second – the stage of the neural network processing. Picture 2 shows a block diagram of this module [3].



Pic. 1 System configuration



Pic. 2 Diagram of module detection and recognition of traffic sign

Let us briefly consider the steps defined in the module of detection and recognition of traffic sign:

Image processing stage:

- preprocessing of the image performs the procedure of selecting objects on the main background (pic. 3).

Pic. 3 Stages of method processing (a) Color map, (b) Balck-White map, (c) Image after Gaussian filter, (d) Image after Gaussian filter and the method Canny

- detection and extraction of the sign is represented in the form of diagram (pic. 4).

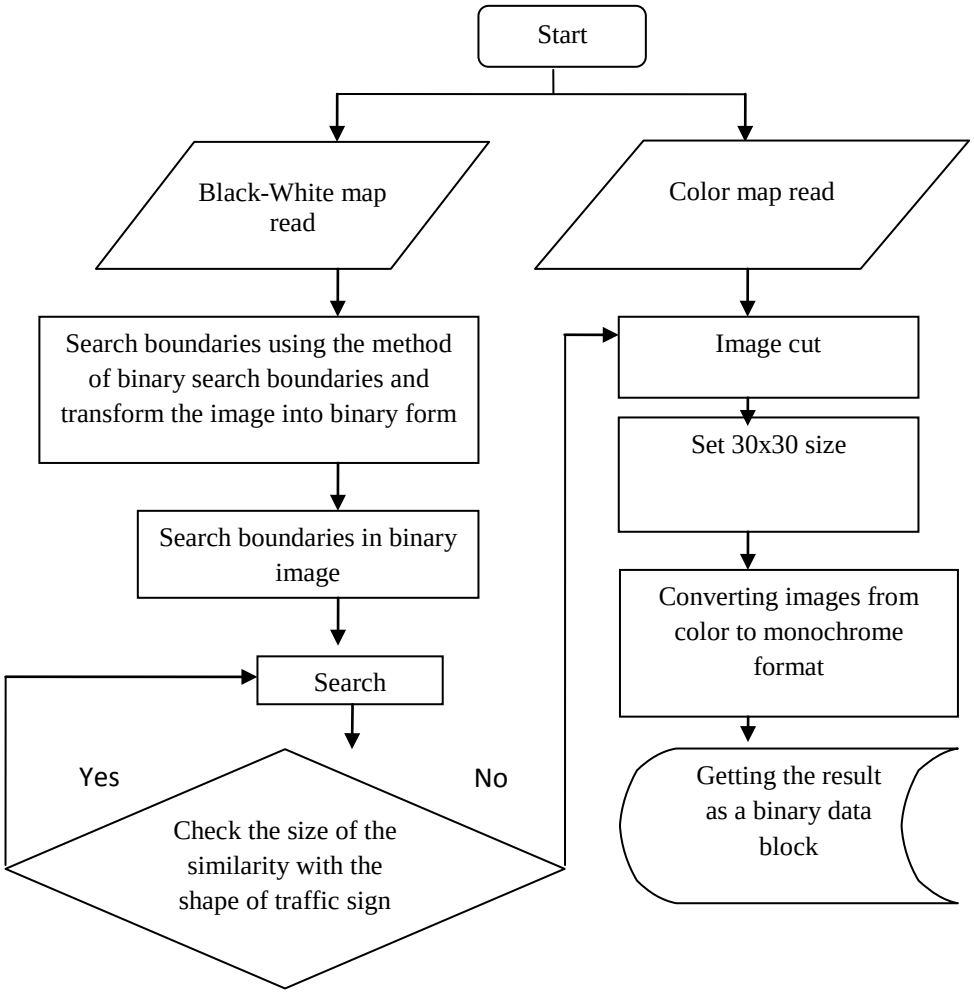
To detect the traffic sign on the image obtained from the video device, a number of filters and fast image processing methods is used:

- prediction technologies;
- Gaussian filter;
- Canny edge detection method;
- method of detecting the boundaries of a geometric object.

Results of the traffic signs detection using this method can be seen in table 1.1:

Table 1.1 Method's results

Summary signs	Signs detection	Partly detection	Result, %
30	21	3	70
100	89	2	89

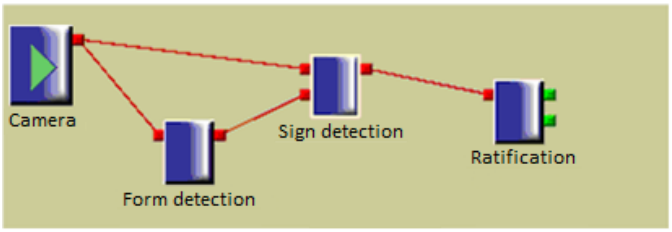


Pic. 4 Diagram of the detection and recognition process of traffic sign

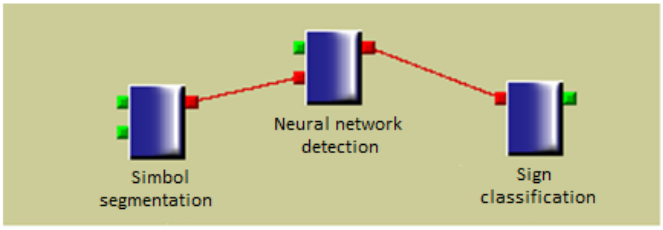
The above technology is currently used by many leading automobile companies in Germany and the USA.

1.2.2 Detection method using analysis of objects features

Also, today a new method is developing and is widely used in the USA, and finding recognition in France. This is a detection method that uses the analysis of object features. The structure of this method has a complex scheme. For example, picture 5 a) shows the general structure method, and picture 5 b) detailed structure of the sign recognition [4].



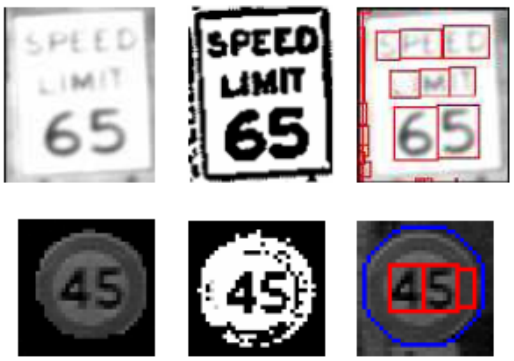
a)



b)

Pic. 5 Structure of method a) General structure b) Detailed structure

The focus of this technology is on selection and recognition of geometric shapes of the signs and traffic signs text data (used mainly for speed limit signs). The technology uses the scale of brightness and contrast, thus reducing the noise on night image processing. Basically, this method widely uses pattern recognition of textual information, contained on a traffic sign, using methods of image segmentation. According to what manufacturers say, it reduces the probability of error in the case of high noise on the processed image. OCR procedure is performed, as shown in picture 6. Furthermore, there is a traffic sign recognition process using the technology of neural network. The network configuration includes a multi-layer network with ten outputs (for the numbers from 0 to 9). An example of the system is shown in pic. 6. Method's interface showed in the picture 7 and the result in table 1.2 [5]:



Pic. 6 Recognition component of the text in the image, use segmentation method

Pic. 7 Results of the method above

Table 1.2 Method's results

Summary signs	Signs detection	Partly detection	Result, %
281	250	2	88

1.2.3 Method based on Scale Invariant Feature Transform technology

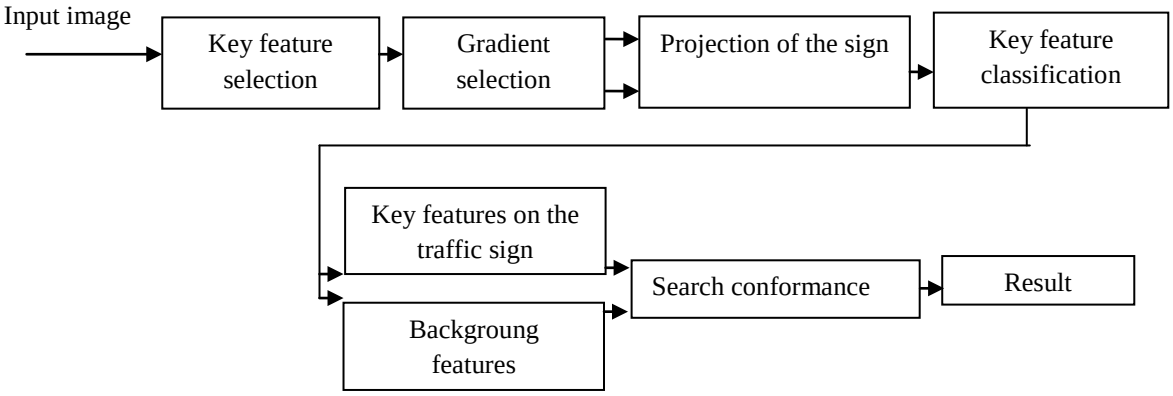
One of the more promising methods for detection and recognition of traffic signs is a technology called Scale Invariant Feature Transform (SIFT)[1]. The main stages of SIFT comprise:

- image analysis and selection of elements that describe the features of the image;

[1] SIFT (Scale Invariant Feature Transform) - geometric transformation of objects with constant characteristics.

- the procedure of recognizing traffic signs does not use a neural network, it is a part of the SIFT algorithm. Thus, using elements obtained describing features of the image in the preceding paragraph, we measure the similarity of a traffic sign on the obtained image with a sample image [7].

The method's structure using SIFT algorithm has the following form shown in picture 8 [7]:



Pic. 8 Structure of a method for traffic signs detection using the SIFT algorithm

This technology has a high resistance to rotation and scale of obtained traffic signs. But in addition to the positive qualities of the technology it is very sensitive to smearing of images and the presence of high noise (table 1.3). In addition, the procedure of processing requires a lot of time and hardware resources [9].

Table 1.3 Results of SIFT

Summary signs	Signs detected	Partly detected
269	230	85

1.3 Findings and statement of the thesis

Summarizing the above methods of allocation and recognition of traffic signs we can define the following positive aspects of existing methods:

- possibility to detect traffic signs in color and black-white images;
- image processing in real time;
- use of neural network technology;
- use of a large number of compression algorithms and filters, such as: Gaussian, the prediction algorithm, Fourier transform, Canny edge, detection contrast and brightness filters, pattern recognition algorithms of geometric figures;
- detection and image recognition technology in the presence of noise, distorted and incomplete picture.

but there are drawbacks:

- it`s complicated to detect and recognize traffic signs in images with high noise level;
- complexity of traffic signs detection in images obtained in the dark;
- difficulty of detection and recognition in images with difficult color range areas (for color images);
- complexity of detection and identification of incomplete traffic signs (for example, if the sign when taking pictures was partially obscured by foreign objects);
- use of intensive and complex methods, filters and processing algorithms.

The described technologies are making it clear that nowadays traffic signs detection and recognition are imperfect and have a number of advantages

and weaknesses that need to be addressed. To achieve this goal, we shall attempt to develop a system that can partially solve some problems described above. The main objectives of the developed system for detection and recognition of road signs can be defined as follows:

- solution of the problem of traffic sign detection on the image taken at night;
- solution of the problem associated with the violation of contrast and brightness of the resulting image;
- using of segmentation technology to detect the type of road sign, instead of the standard technology of neural network;
- reduce the computational complexity of procedures of detecting and recognizing road signs.

2. Development of traffic signs detection

2.1 Methods description

The task of traffic signs detection while driving the vehicle is quite topical at the moment because of the increased number of emergency situations. The main causes are inattention, and drivers not observing the speed limits on the road. To solve these problems it is necessary to develop a method which enables us to detect the traffic signs automatically and warn the driver.

The developed method is a multilevel system based on several processing stages of the data, on the basis of which the objects, i.e. traffic signs, will be analyzed.

Since this thesis does not imply the using of expensive equipment in an effort to reduce the cost of technology, but without decreasing speed, the choice is between video capture devices with CCD or CMOS technology [11].

In this thesis, for video capture we will use an external device - a webcam. This choice is made for the small cost of equipment, but the webcam has a number of advantages, such as: auto focus, manual adjustment of the frame frequency per second as well as compatibility with a PC that will facilitate a quick setup of the device to work with the software.

After the setup of connect is to external video capture device, there will be procedures for receiving the video stream with a certain frequency. For the developed method we should receive static images (frames) with a fixed frequency.

After this procedure the image will be transferred for processing [12].

The block of preprocessing performs image processing for its restoration. Typically, this type of system is a set of a five-level processing. They include the following stages of primary processing [13]:

- smoothing of the image (deconvolution, spatial filtering);

- partitioning the image into fragments with subsequent filtration;
- apodization for increasing the resolution;
- expansion of image boundaries;
- superresolution.

In this thesis, it is expedient to use only the first two stages, because the method is not in need of full restoration of the original image.

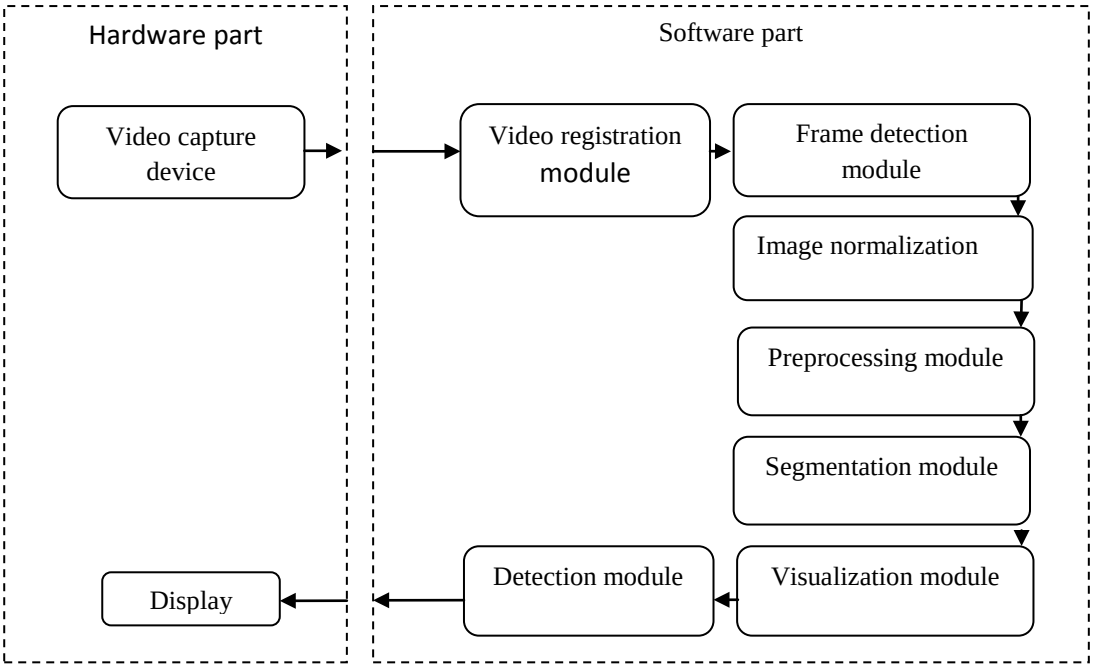
After the initial processing, the reconstructed image goes to the next stage of processing for the procedure of segmentation and morphological analysis. This procedure is provided to find the desired object in the image (a traffic sign) and implement a series of operations for its selection in minimum time.

At the last stage the object is displayed on the display.

2.2 Method specification

There are many methods and systems today that deal with traffic sign detection and recognition. Many methods today have almost the same kind of structure. Thus, we can say that existing methods and systems have an optimal structure and a substantial modification of this model is not required for the development of a new, more efficient system or method.

The developed method in this thesis can be represented as a set of modules. Each module carries a certain number of operations and transfers the result to the next module. The structure is divided into two parts: a hardware and a software one, and has the following form (pic. 9).



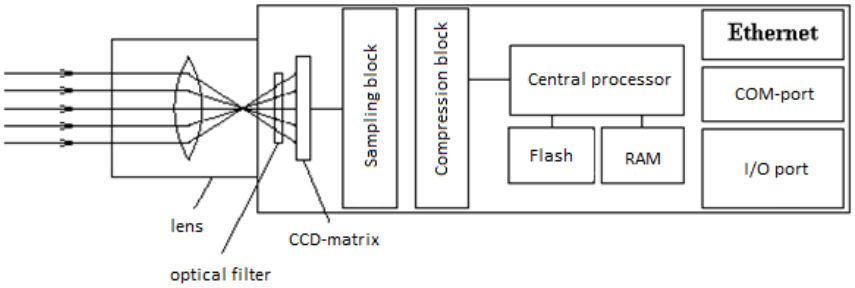
Pic.9 Diagram of detection system of road signs

Video capture device:

A webcam is a digital device that captures video and converts the analog video to a digital form, compresses a digital video and transfers the video images by a computer network. Therefore, the web camera includes the following components (pic. 10) [14]:

- CCD (CMOS) - matrix;
- Lens;
- Optical filter;
- Video capture card;
- Block compression video;
- CPU and the embedded web server;
- RAM;
- Flash memory;

- Network interface;
- Serial ports;
- Warning inputs / outputs.



Pic. 10 Webcam diagram

When choosing a web camera it is important to choose correctly the way it connects to the computer. Most models are equipped with a USB-connector, but, as we know, there are two versions of this interface, and they differ greatly in speed. If your camera supports the version of USB 1.1, then the video quality is not the best: 640x480 dots (pixels) at 15 frames per second (or the frame rate 15 fps). Frequency of the frame must not less than 30 fps (now almost all webcams have a similar frequency). USB 1.1 can not use a large data stream. Therefore, we recommend to choose a webcam that supports the USB 2.0 [15].

This interface has several advantages, primarily due to the possibility of a "hot" connection to a personal computer (PC). This is one of the major pluses. This interface has a higher bandwidth than USB 1.1, and is 480 Mbps. It's based on the method of dividing one frame with a duration of 1 ms. Thus, this specification uses a smaller volume of buffer [16].

As it is known there are two main types of matrices, and they vary in a way of reading information from the sensor. In matrix-type CCD information is read from the cells in series, so the processing of the file can take much time.

They are relatively cheap, and besides, the noise level is less in their images. In matrix-type CMOS the information is read out individually from each cell [15].

Besides the sensor technology, it is necessary to consider such an important parameter as resolution. The quality of transmitted images depends on the matrix size. The range of values lies in the range of 0.1 to 2 megapixels (MP), but the most popular resolution VGA amounts to 0.3 megapixels. For online video conferencing a resolution of 320x240 pixels is enough. Standard 640x480 pixels are ideal for recording small video [15].

RGB - the color representation in the coordinates space of the additive mixture, which describes the color as a sum of three basic colors: red (Red), green (Green) and blue (Blue). RGB is represented as a three-dimensional coordinate system. Each coordinate represents the contribution of each component in the resulting color ranges from zero to its maximum value. The result is a cube, forming a color space RGB.

YUV - color is represented as 3 components - brightness (Y) and two chroma (U and V). This model is widely used in TV broadcasting and storage / processing of video data. The brightness component contains a "black and white" (grayscale) image, while remained two components contain the information required to restore the color [17].

Thus, on the basis of the above capabilities of software and hardware webcam, you can optimally configure the video capture.

Capturing video will be produced with a resolution of 640x480 (VGA), because a lower resolution will result in loss of detail. This can lead to deterioration of the video capture and increases the probability to miss analyzed frames. Moreover, the data will not be transferred by channels with low bandwidth. Higher resolution is not supported by this device.

The color model, which is used in this method, is RGB. It is the most effective for the transfer, storage and video capture. This model has a broader

range of methods for processing video images, compared with the YUV-model. The color depth will be 24 bits (Truecolor). It is believed that this depth is the minimum to create photorealistic images with half-tones, so that the eye sees no sharp boundaries in the transition from one color to another.

After setting up the video capture device, the process of transferring video to the program module - starts. This module makes the procedure of capture in a cyclic form and the procedure for termination of data acquisition.

After the procedure of retrieval of the video stream, it is necessary to carry out initial processing of the image.

As stated above, the initial processing of this method excludes the use of stages: apodization, the expansion of borders and superresolution. As a result, only a smoothing procedure remains. Additional steps of image normalization, are required for images obtained in the night. Also you need to use some techniques to improve digital images, such as: increased contrast, correction of dynamic range, the difference method, threshold processing of color images, etc. Proceeding from the above-described, the choice of filters and techniques was done as:

- forming the normalized images based on images with low brightness;
- wiener filtering;
- improving the quality of the image using the method of sharpening.

Let us consider in more detail the methods described above.

The first stage, after obtaining an image from an external video capture device, is the formation of the normalized images based on images with low brightness.

Traditionally, to solve the problem of modeling normalized images a standard algorithm is used, which consists of the following steps:

- total contrast enhancement;
- total increase in brightness;
- edge detection;
- decrease in shades of blue.

First we need to decompose the resulting image into its component parts. Thus, the first step is to analyze the intensity of color components, and if it is in a limited range, it is necessary to apply stretching to the dynamic range of intensities of color components [20].

If the intensity of the pixels is concentrated in a narrow dynamic range, then you should use the method of extending the dynamic range of intensities of pixels. The basic principle is to stabilize the color increasing the red, green and blue. It should also introduce a shift of the inflection point of the curve up or down, but strict rules for the operation do not exist, so the linear equalization is optimal (pic. 11)[21].

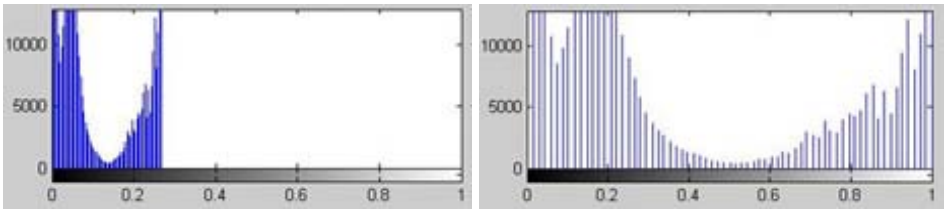
These changes according to expression (2.2.1):

$$L_{res} = \frac{L - L_{min}}{L_{max} - L_{min}}; \quad (2.2.1)$$

where L_{res}, L – array of the source and the resulting images; L_{min}, L_{max} – minimum and maximum values of the original image.

Also note that the transformation (2.2.1) is effective when the intensity of pixels in the image is concentrated in a narrow dynamic range. If we apply these transformations to an image with a normal distribution, then the desired effect is achieved because the color components histogram occupy possible range [21].

The improvement of visual quality of the original image can be achieved if the modified expression (2.2.1), raising the expression to the power α, where α - coefficient of nonlinearity [21].



a) b)

Pic. 11Image normalization a) brightness of the image before normalization; b) brightness of the image after normalization.

A major shortcoming of the resulting image in night is a low contrast. This explains the poor visibility of fine details and low visual quality. To overcome this limitation it is necessary to improve the contrast of images. This operation is carried out separately for each color component. This gives a good result, but it is even better to carry out the operation for all components, that will bypass the problem of colors unbalance [22, Chap. 11.6.5].

After normalization of the image obtained at night, we proceed to the next stage of the primary image processing. It is necessary to conduct the operation of Wiener filtration to eliminate noise in the image.

The optimal Wiener filter can filter an image so that if closes to corresponds to the truth, provided that the spectrum of image and noise are known a priori. As we know, inverse filtering takes a low noise immunity, because it does not take image noise. Much less sensitive to noise and zeros of the transfer function is a Wiener filter, which uses information about the spectral power density of the image and noise [23].

The spectral density of the signal can be represented by the following expression (2.2.2):

$$S_{II}(v) = F[R(\omega)]; \quad (2.2.2)$$

where $R(\omega) = \int I(x) * I(x-\omega)dx$ – autocorrelation function.

The cross-spectral density is represented by the expression (2.2.3):

$$S_{II'}(v) = F[R(\omega)]; \quad (2.2.3)$$

where $R(\omega) = \int I(x) * I'(x-\omega)dx$ – cross-correlation function.

In constructing of the Wiener filter we need to minimize the standard deviation of the processed image from the object. Then this mean takes the form (2.2.4):

$$E\{[I(x,y) - I'(x,y)]^2\} = min; \quad (2.2.4)$$

Transforming these expressions can be shown that the minimum is achieved when the transfer function is given by (2.2.5):

$$D(v_x, v_y) = \frac{S_{II'}(v_x, v_y)}{S_{II}(v_x, v_y)}; \quad (2.2.5)$$

Then the image reconstruction will be done by a filter that has the following form (2.2.6):

$$R(v_x, v_y) = \frac{1}{D(v_x, v_y)} * \frac{|D(v_x, v_y)|^2}{|D(v_x, v_y)|^2 + D(v_x, v_y)}; \quad (2.2.6)$$

If the image does not have noise then the spectral density function of noise is equal to zero and the Wiener filter becomes a normal inverse filter. Wiener filter much better suppresses noise. Oscillating noise on the results of image reconstruction is due to boundary effects. Noise level is substantially less than with the inverse filtering, but Wiener filter partially compensates for edge effects, which make poor quality of recovery (pic. 12)[23].

After the procedure of image reconstruction using Wiener filtering, smoothing operations and brightness normalization we can proceed to the next step. At this stage, there will be edge detection on the image (pic. 13). The proposed method is the easiest to implement and includes the following steps:

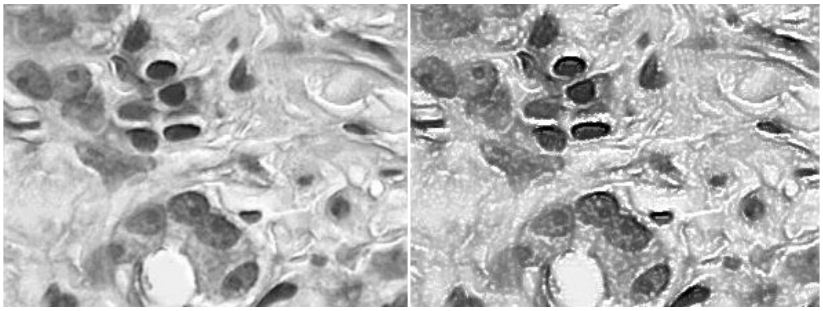
- read the images separately for each component of the RGB;
- calculation of the mean for each component;
- displacement of each of the values of the components on the mean of the expectation by the following formula (2.2.7):

$$x_i = M + (x_i - M) * k * 50; \quad (2.2.7)$$

where: M - calculated expectation for each component; k - the noise factor Gaussian by Wiener filter resulting.

Pic. 12 Restoration of distorted images using the method of Wiener filtering.

Increased sharpness of objects increases image quality of any method of contour segmentation, but also may partially reduce the accuracy of edge detection of image objects. With a signal to noise ratio <5, this method has shown excellent results, but in the opposite case, when> 5, is more effective the method of contour segmentation based on wavelet transform (RWT) [24].



a) b)

Pic. 13 Edge detection process a) Original image before applying the sharpening method; b) Image after applying the sharpening method.

As a result of all operations we obtain a normalized image, smoothed by Wiener filtering and edge detections of image elements. Thus, this procedure will increase the probability of detection of the desired object (traffic sign) in the image. Also it will allow us more accurately and effectively determine the position of the object in the image and its type.

After the image has passed the primary filter, we can go to the stage of desired object detection in the image. Currently, there are many methods and technologies to get the desired result, but the most common is the method of segmentation.

Among the wide variety of different methods of segmentation, we can identify the most effective:

- image segmentation based on the clustering method k-means;
- segmentation method of controlled watershed;
- texture segmentation using texture filters.

The most suitable - the method of controlled watershed segmentation.

The separation of touching objects in the picture is one of the important tasks of image processing. To solve this problem it is necessary to use the marker watershed. We need to highlight the "catchment basins" and "dividing

line" on the image by processing the local regions according to their brightness characteristics [28].

The method of marker watershed is one of the most effective methods of image segmentation. This method includes the following basic procedures [28]:

- calculation of markers of the foreground. They are calculated on the basis of connectivity analysis of each pixel of the object;
- calculation of background markers. They represent the pixels that are not parts of objects;
- modification of segmentation functions, and the basis of the values of the location markers of the background and foreground markers;
- calculation based on the modified segmentation function.

Segmentation procedure can be divided into several consecutive stages [25]:

- reading the image;
- morphological expansion of images;
- color filtering;
- filling gaps;
- filling of internal gaps;
- smoothing of the object;
- detection of objects by geometric shape.

Let us consider these steps in more detail.

The first phase will be skipped because it does not have any serious difficulties in implementation.

The application of morphological operations expansion for contours, using a gradient mask gives acceptable results, which is to fill the holes in the middle of the object, thereby reducing the number of small elements and noise in the image. (pic. 14). [25].

a) b)

Pic. 14 Morphological expansion of the image a) to expand operations b) after operation

The next step is a filtering of color. At this stage the original image is converted to a binary form. In this case the desired colors are replaced with the value "1" (white), and the rest of the background is set to "0" (black). Thus image contains only white objects on black background. The operation consists of a filter cycle by the given condition. The condition has a limit which corresponds to a particular color, in case the pixel stay in the prescribed limits, it is replaced with the value "1", otherwise to "0" (pic. 15).

Pic. 15 Color filtering operation

The next stage is detecting the entire setup. On the condition that the image can represent more than one object it is likely that only one of them will be seen . The challenge is to extract the object. This task is to ensure that all binary objects must be detected and then calculate their ratio and convexity. Based on these data and a priori data of the desired objects, you can extract the necessary facilities [25].

Let N be the number of pixels belonging to the object. The entire set of pixels p (x, y) belonging to an object, will he denoted by Q. Then the coordinates of the center of mass of the object are calculated as (2.2.8) [25]:

$$x_c = \frac{1}{N}; y_c = \frac{1}{N}; npu \sum_{p(x,y)\in\Omega} x; \sum_{p(x,y)\in\Omega} y; \quad (2.2.8)$$

It is necessary to calculate additional values for the calculation of other characteristics. The equations below are used for calculating the central vectors (2.2.9, 2.2.10) and their center of mass (2.2.11):

$$U_x = \frac{1}{12} + \frac{1}{N}\sum (x - x_c)^2; \quad (2.2.9)$$

$$U_y = \frac{1}{12} + \frac{1}{N}\sum (y - y_c)^2; \quad (2.2.10)$$

$$C = \sqrt{(U_x - U_y)^2 + 4U_{xy}^2}\ ; \qquad (2.2.11)$$

Then the length of the maximum $A_{\max}$ and minimum $A_{\min}$ axes of inertia are calculated as (2.2.12, 2.2.13):

$$A_{\max} = 2\sqrt{2} * \sqrt{U_x + U_y + C};; \qquad (2.2.12)$$

$$A_{\min} = 2\sqrt{2} * \sqrt{U_x + U_y - C};; \qquad (2.2.13)$$

The lengths of the main axes of inertia used for calculating the eccentricity and orientation of the object. The eccentricity is defined by relation (2.2.14):

$$E = \frac{2\sqrt{(0.5 * A_{\max})^2 - (0.5 * A_{\min})^2}}{A_{\max}}; \qquad (2.2.14)$$

Via the above parameters, you can create a number of characteristics of geometric objects detected in the image. Thus, these parameters allow you to select the desired objects from among the total number.

After the recognizing of the object on the image the result will be displayed on a display. This part is relatively simple. Therefore the description of this part is not required.

2.3 Logical structure of method

Any logical structure includes a standard set of elements. They can be summarized as follows:

- Initialization of the subject;
- Initialize the object;

- The subject of analysis;
- Analysis tools;
- Methods of work;
- The result of analysis.

Let us consider in more detail this structure.

On the first place there is the subject of initialization. The subject is a dynamic image, because of the vehicle is moving. The image of the road includes the border area, which usually has objects such as road signs. Since the basic number of signs is located on the right side of the highway then the capturing should be on the right side.

The Object of this thesis should be considered as a software-hardware system. Hardware does not have any complex elements in its structure and has only a video capture device - web camera and output device i.e. a display. The software part is a software in a form of an executable file that captures video, its analysis, frame extraction and analysis, with the result on the display. The Software part includes a set of techniques to process captured data more effectively.

The subject of analysis in this case will be a frame. Later this frame will be processed and analyzed to detect a traffic sign.

The sources code of analysis includes a set of packages or devices thatare compatible with the software development environment. In this case, the analysis tools must include the following tools:

- video capture package and video processing flow;
- tools package and techniques for primary processing of images;
- segmentation package;

- tools package for analysis, detection and detection of objects;
- input/output data.

The methods may include such means as:

- preprocessing filters (Canny, a method of smoothing and threshold processing, normalization filter, method of equalizing the luminance component and contrast, etc.);
- segmentation methods (Image segmentation based on the clustering method k-means, segmentation method of controlled watershed, texture segmentation using texture filters);
- methods of object detection on the primary image (the coefficient of correlation and on the basis)

The result of analysis is the detection of the object in the form of a traffic sign, if it is present in the image. And display output of the object.

2.4 Closing

The developed method of traffic signs detection in the video stream is designed to improve the detection technology and accelerate it. In contrast to existing methods this method has several advantages, such as:

- if work in daylight and at night time. Thus is possible with the introduction of the normalization method which allows for regulation by the use of brightness and contrast filters that convert the image to the normalized form;
- also a distinguishing feature of this method is a segmentation method, which is used in conjunction with the method of analyzing the properties of the object and allows one to detect the desired object in the image. In contrast to this method other methods use a neural network

with learning or without learning. It should be understood that the use of neural networks is sometimes more difficult.

Otherwise, the method of object detection has not undergone any fundamental modifications. The method also realizes primary processing which is responsible for image smoothing and noise reduction. Also saved mechanisms to capture video stream and output the result. The disadvantage of this method includes the fact that it does not work in real time.

3. Software development

3.1 Selection of programming language

As this work is devoted to the development of traffic sign detection method so for the effective creation of software we must choose the optimal environment for its development. The most suitable option in this case is MATLAB.

MATLAB System is a comprehensive application for research. This system is used not only for the development of new equipment, algorithms, software but also to modify and improve existing ones. The main advantage of MATLAB is the relative ease of manipulation with any types of data, as well as convenient means of combination of all possible schedules. Besides the described advantages MATLAB has a complete help file, which contains not only a description of functions, but also a large number of examples allowing as to fully understand the operation of certain functions.

Speaking on the mathematical aspects of MATLAB, it should be noted that its notation is very close to those which have been used for long in mathematics, and it is noticeably simplifies the development of numerous mathematical commands.

These advantages are justified by the fact that MATLAB has a number of advantages including:

- built-in image processing software Image Processing Toolbox. This package has several advantages such as [1]:
    - high speed;
    - an ability to view images, including image viewing metadata, zoom and view large images;

- new features for processing colored images, which give a possibility to analyze image and make colored space conversion;
- a number of new methods of image enhancement including methods for contrast transformation, adaptive transform histogram and stretching;
- the development of a new method for tracking the edge of image objects is easy;
- improved speed of calculation and image filtering;
- optimized speed and memory used by other functions of the package.

- Built-in package Image Acquisition Toolbox, which is a combination of the following functions [1]:
  - access to image data directly from MATLAB;
  - interaction between the most modern standard analog and digital devices to capture images;
  - interaction between Windows-compatible video devices, Web camera, USB and FireWire (IEEE-1394) camcorder capture card video, digital (DV) camcorder);
  - an ability to view video, support non-standard and standard video formats, including CCIR, NTSC, PAL, RGB, RS170, SECAM, and S-Video;
  - simultaneous capture and image processing, access to the functions supported by hardware, such as gain, brightness, contrast, and the choice of synchronization.

Thus MATLAB allows without any problems to ensure not only the capture of video from an external device, but also the processing of the data.

3.2 Program structure description

The structure of the developed method includes a set of individual files, which represent individual functions. Each new function is linked, usually with a specific module of the developed method. Let us consider the hierarchy of files and what they comprise.

Let us consider the work file in more detail.

Function (Appendix, Main.m) has no input/output arguments, and is the most important in the hierarchy of software files.

The first step is to conduct the operation of memory clearing so that the data of previous launches, or other unnecessary data will not affect the work of the current components. This is a standard command in MATLAB - «clear all;».

After you are sure that all bad data that can impede the normal operation of the program is excluded, you need to call the function «VidCapInfo». This function will provide full information on existing and available video capture devices. More detailed work of this function is considered in the description of the file «VidCapInfo.m».

Next there is a function «VidSetup» (Appendix, VidSetup.m). It is based on the data obtained on the video capture devices, allowing us to initialize one of them. In the future we will get data for further processing and analysis by using this function. The structure of this function is considered in the file «VidSetup.m».

After the device initialization procedure, you can start the process of capturing video stream and processing. This operation is represented as a cycle of a certain number of operations. The start of the video capture process should always be the operation «start (WebCam);», and should finish by the operation «stop (WebCam);». Here, as a parameter of both functions is the structure «WebCam» which contains information about the video capture device, which

was initialized. Operation «start» and «stop» are included in the package Image Acquisition Toolbox environment MATLAB (Appendix, Main.m).

Before the beginning of the operation of capturing a video stream from the device and its further processing, we must initialize the parameter «times», which is responsible for the number of cycles to access device memory. This will limit the time of the program.

Now we can proceed to video capture and treatment. These operations are carried out in a loop of the form «while (WebCam.FramesAcquired <times) ... end;», where «WebCam.FramesAcquired» - captured number of frames (Annex, Main.m).

Next function - «trigger», directly switches the device to the capture mode.

Once the operation of capture has finished, we can now proceed to the reading of the data. This operation must be carried out from the memory of the video capture device and transformed into a sequence of frames. The function «getdata» is responsible for the process of reading frames from the memory (Appendix, Main.m). The result of the function «getdata» appears in the form [32].

The Function «getdata» returns MATLAB data that is using colored space, determined by the properties ReturnedColorSpace [32].

Once a frame is received from the memory of the capture device, a copy should be saved in order to identity the areas that are suspected to contain information about the desired object, i.e. locations of traffic signs at the end of the processing and analysis.

Processing of the received frame begins with the primary treatment. As described in the specification of the program in the previous chapter the primary processing contains the steps of image normalization and methods of

smoothing the image with a subsequent increase in sharpness. As seen from the loop primary processing contains:

- «NightImProc» - function performs normalization of the image captured is low light;
- «WienerF» - function performs Wiener filtering, which reduces the noise in the image;
- «EdgeDt» - function sharpens the borders of objects in the image.

After the preprocessing, you can proceed to the detection of the object. For this we have implemented two main functions: «ObjDt» and «Detect». The first of these functions includes the set of filters that allow you to provide the necessary facilities for their color and shape, while the second deals with the allocation of detected areas in the original image. Thus the screen will display the final result of the developed method.

The first one is a function of obtaining information about the existing capture devices:

As you can see, this function (Appendix, VidCapInfo.m) has no input/output parameters. It is intended only for choosing devices of video capture. Referring to this function, the screen will display all information about the video capture devices.

Operation «imaqhwinfo» (Appendix, VidCapInfo.m) is designed to obtain information on embedded devices capture [33].

Thanks to the information received, you can tune the necessary equipment for further processing with it.

For this there is the next function - «VidSetup» (Appendix, VidSetup.m). Based on the information it allows to initialize the device. The function has no output parameters, but has input, which sets the basic settings.

Object properties «VideoFormat» (Appendix, VidSetup.m) provide the given format. The function selects a given video object and a description of its name in the properties SelectedSourceName. To access the video object, which is used in the capture, use the function «getselectedsource» (Appendix, VidSetup.m) [34].

Instead of device name its identifier deviceID is used. When multiple devices have the same name the first available device is used. This function is based on the initial data and generates device configuration files «devicefilename» in the form of video format [34].

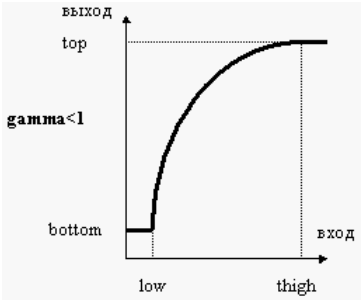
Now, after initializing the device you can begin the preprocessing of captured frames.

The first of these functions - «NightImProc» (Appendix, NightImProc.m). Has one input argument and one output. As an input argument the function takes the captured image and gives the normalized output image:

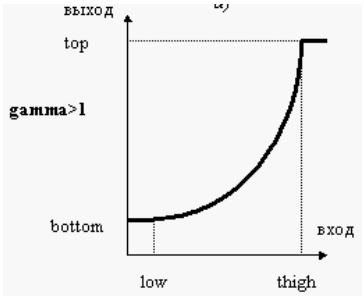
If the schedule of brightness distribution has an upper limit that is lower than half the possible range, it is necessary to apply the normalization.

Consider the case when the condition above is satisfied. The function «imadjust» is used to stretch the range of brightness (Appendix, NightImProc.m). It needs to be applied on each component of source image, in order to improve quality. This function has the syntax «imadjust (inImg, [low high], [bottom top], gamma)», and creates grayscale image output by contrasting the original grayscale image. Using gamma index you can also perform the conversion called gamma – correction (pic. 16) [35].

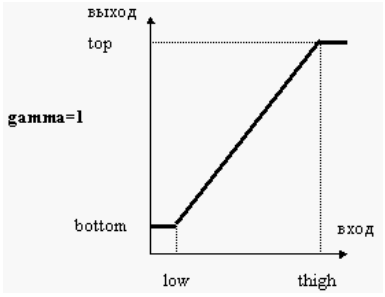
However, there are classes of images for which the linear or nonlinear transformation of brightness ranges is not always effective. Especially when the image brightness occupies the maximum possible range. That is why the normalization is used only selectively [35].



a)



b)



c)

Pic. 16 Characteristics of transmission levels for different values of the gamma a)gamma<1; b)gamma>1; c)gamma=1

After the previous operation, image sharpness should be improved by using the function «imfilter». For this purpose, firstly you need to initialize a mask of predefined type «fspecial» (Appendix, NightImProc.m). This function will return the mask on a two-dimensional linear filter defined as the first parameter.

Now consider the function «imfilter» (Annex, NightImProc.m). The function filters the multidimensional input array. Input array must be a non-sparse numeric array of any size and dimension. The resulting output array has the same dimension and the data format as the input array [37].

As a result of the best filter settings the sharpness of borders is increased . The result of the function is demonstrated below (pic. 17).

a)

b)

Pic. 17 Normalization operation a) Original image; b) Image after normalization

After this procedure, the resulting image still lacks sharpness and has a high noise level, thus can not fully guarantee the success of object detection. As a result, it is necessary to make some additional filtering to improve image quality.

The first filtration - Wiener filtration «WienerF» (Appendix, WienerF.m):

The function has one input argument which accepts the input of the original image for processing, as well as two output arguments. The first of these two arguments sends the image after processing the Wiener filter and the second argument passes the value of noise, which is calculated during the Wiener filtering (Appendix, WienerF.m).

For more effective filtration we need to add value image noise. Since this information is largely a priori, there is no standard methods for calculating this

value. It was therefore decided to use the standard deviation. For this we use the function «std» (Appendix, WienerF.m), in relation to one of the color components of the image.

After the calculation of the noise, you can go to the filtration. For this the function «wiener2» is used. Function «wiener2» (Appendix, WienerF.m) creates an image, which is the result of an adaptive Wiener filtering of the original image (pic. 18). The parameters of this function sets the size of a sliding window within which the mean and standard deviation of brightness values is estimated. The noise parameter, which was calculated above, sets the power of the Gaussian noise, which use damaged the image [38].

Pic. 18 Wiener filtration

Thus, the output image has reduced noise level and a normalized parameter of brightness. We need only to extract the edges of the image, then to go to the segmentation and detection module.

Boundaries enhancement is carried out with the following function «EdgeDt» (Appendix, EdgeDt.m):

Input parameters contain the image filtering as well as the ratio of the Gaussian noise, which is calculated in the previous function «WienerF» on the basis of the data and the generated output image with an increased level of sharpness of the borders of image objects (Application, EdgeDt.m) (pic. 19).

Pic. 19 Edge detection

The preprocessing of the resulting image ends in this step. After the procedure of image reconstruction comes segmentation of the reconstructed image with additional filtering. All these procedures are necessary in order to cut out unnecessary information and for more efficient detection of the location of the object.

Segmentation begins with the function «ObjDt» (Appendix, ObjDt.m), but firstly the function «ColorDt» (Appendix, ColorDt.m) is used. Let us consider this process in more detail.

For understanding the method, it is necessary to consider functions in a different order. Firstly the function «ColorDt» (Appendix, ColorDt.m) is used.

The input function takes two parameters. The first of them is an image after preprocessing, and the second is a color index. The image will be filtered by the second argument. Since the investigated area contains objects of three primary colors: yellow, blue, red, filtering will be done on them. The main idea is to extract a color and give the output information in a binary image on which the desired color appears white and other colors black. This technology is a method of segmentation on the basis of a managed watershed. That is why at the first level of processing morphological analysis of the image is performed.

This analysis is marking objects. To mark the foreground objects different procedures may be used. In this example, we will use morphological

technology, called "disclosure through rehabilitation", and "closures through rehabilitation." These operations allow you to analyze the inner region of image objects. Firstly we use the operation of disclosure using the function «imerode» and «imreconstruct» (Appendix, ColorDt.m).

Function «imerode» operates thinning of binary or packed binary image and returns a refined image. The second argument is a structural element of the object or an array of structural elements returned by the function «strel» (Appendix, ColorDt.m) [39].

Function «strel» creates a structural element of a type described in the first parameter. Depending on the shape parameter input parameter, a structural element «strel» may have a number of additional parameters [41].

Function «imreconstruct» (Appendix, ColorDt.m) performs morphological reconstruction of the input image by the image mask. Input image and the mask should be two grayscale or binary images of the same size. The returned image is a halftone or binary image, respectively. The elements of the input image must be less or equal in value then the corresponding elements of the array mask. By default, the function «imreconstruct» uses 8-connected neighborhood for two-dimensional images and 26-connected for three-dimensional images (pic. 20) [40].

Pic. 20 Morphological analysis result

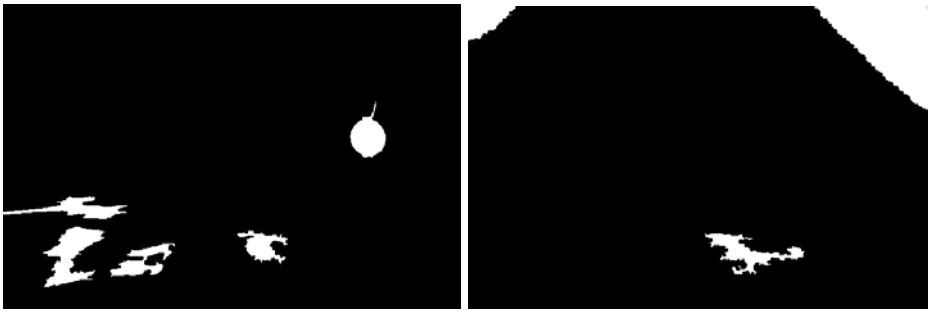
After the morphological operations we must go on to filtering by color. This function has three identical processing steps for each of the basic colors, respectively. The difference between the data blocks is only in the fact that they operate at various intervals of color. The result is a binary image in which the white color represents the target color.

The final step in color filtering is to perform the function «DelNoise» (Appendix, DelNoise.m), the input of which receives the filtered image. Let us consider it in more detail.

Function «DelNoise» carries an additional number of operations of morphological analysis, which are used in the method of controlled watershed. The function includes the following operations:

- «Imfill» - this function performs the filling of the original binary image (Appendix, DelNoise.m);
- «Imdilate» - performs an operation of dilation on a grayscale, binary or packed binary image, returning as an output a modified image. The second argument is a structural element of an object or an array of structural elements of the object returned by the function «strel» (Appendix, DelNoise.m).
- «Bwareaopen» - function removes from the binary images all connected components (objects), which area is less than the specified limit of pixels and places the result in the output image processing (Appendix, DelNoise.m).

Thus, after filtering by color and removal of unwanted objects from pictures, we get a binary image that contains a number of objects of different shapes. Among these objects, we can assume that there are also unknown objects (pic. 21).



a) b)

c)

Pic. 21 Color filtration a) red-filtration b) blue-filtration c) yellow-filtration

For their isolation it is not enough to make filtering by color, but we need to analyze the geometric shape of objects for more efficient detection. To do this we apply the function «ObjDt» (Appendix, ObjDt.m). As a result of this function we obtain a structure that contains the selected areas.

To identify the region of desired object, you need to consider all objects that are located on the image and store the coordinates of only those objects that fall under a number of properties. For this purpose, the function «ObjDt» contains a set of functions:

- «Imfeature» - function calculates the signs of all objects identified in the matrix of numbers of objects (Appendix, ObjDt.m). The elements of this matrix, having a value of 1, refer to the first object, having a value of 2 belong to the second object, etc. If it is 0, then it is background. The values of the attributes are returned in an array of structures. The number of array elements is equal to the maximum element of the

matrix and, consequently, the number of objects in the image. Each structure contains the attributes for a single object [41].

In this particular case we use only the options:

- «BoundingBox» - bounding box (x, y, width, height);
- «Centroid» - center of mass of the object;
- «Extent» - fill factor: is the ratio of the area of the object to the area of the bounding box;
- «Eccentricity» - the eccentricity of an ellipse with the principal moments of inertia, equal to the principal moments of inertia of the object.

These features are determined and provide the necessary objects in the image.

Thus, the final data is generated in the form of structure coordinates of each detected object and we can proceed to the final operation of allocation of the detected objects in the image. This will demonstrate the working of the method and is likely to reveal the advantages and disadvantages.

This task is performed by the function «Detect» (Appendix, Detect.m), which, with incoming information in it as a set of object coordinates, allocates the detected objects with rectangular windows, thus visualizing the procedure as shown (pic. 22).

Pic. 22 Presentation of processing results

At this stage, the work of the method is completed. After the above procedures, the program performs iteration. Thus, we obtain a loop of detection procedures. At each new iteration of the device reads the captured image from memory.

3.3 Organization process of collection, transmission, processing and issuance of data

Collection of information is a process of retrieval and analysis of information in the method, which serves as processing of data.

The purpose of the collection is the preparation of information to further processing in the information process. As this phase the circulation cycle of information begins and it is very important, because the quality of its performance depends on the quality of information that will be used in the method.

The main stages of data collection include the following:

- initial perception of information - the procedure is an allocation of the basic properties coming onto the input of information and their improvement. Since at the input this method receives a video stream, the main criteria here are: refresh rate, resolution of the resulting video stream, as well as the possibility of auto focus and automatic correction of brightness and contrast. From the specs video capture described in Chapter 1, we can say that it has the optimal set of parameters;

- provided use of classification methods. These are: segmentation and morphological analysis;

- recognition of unknown objects means detecting the whole set of objects in the image and classifying them by color and form that allows you to fully describe the accuracy of the results;

- display the result of work done by the method of visualization of the original image and the overlay of the boundaries that were discovered during the operation and fulfill the criteria of selection.

Transmission of the information is carried out directly by the two main channels: the USB capture device to the program shell, and the result of the work being transferred to the device for rendering, i.e. on the screen. Other processes occur in the program shell, which reduces the risk of data loss when transferring from one module to another.

3.4 Experimental studies of the method

Experimental studies of the method were conducted in different conditions.

Let us consider one of the experiments.

As described in the specification of the method, the capturing on the video stream must be carried out with the capture device. The resolution of captured video must be 640x480. When all the necessary parameters were set, and the processing run, we obtain the following results.

- video stream capture, and read from the memory of the capture devices a frame for analysis (pic. 23);

Pic. 23 Primary image

- analyze the image for the condition of low luminance. The results show that the image does not require normalization, which means you can go to the primary processing;
- first, at the stage of preprocessing decrease of the noise, using Wiener filter. Result of noise reduction leads to the fact that the image is missing small parts. This is due to the fact that the Wiener filter smooths the image (pic. 24);
- as the Wiener filter leads to smoothing, the boundaries between objects in the image are diluted. This can lead to a situation that the objects will be hardly visible. To solve this problem in the structure of the method we include a stage of edge detection of objects in the image (pic. 25);

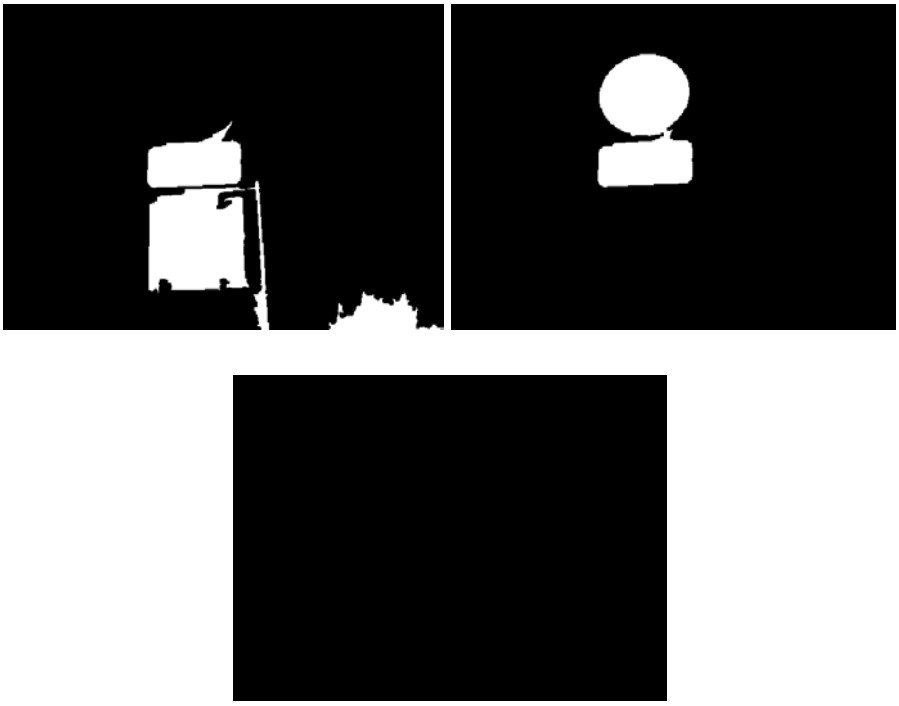
Pic. 24 Wiener filtration result

Pic. 25 Edge detection result

- after preprocessing has been done, there is a morphological analysis. Morphological analysis is to perform multiple operations. Firstly, the operation marking foreground objects, which reduces erosion and stretching of the image. This leads to the fact that small items disappear, and only major ones remain. Secondly, this operation is a morphological reconstruction, which changes the image to the mask, after its reconstruction (pic. 26);

Pic. 26 Morphological operations result

- now there is a filtration by color, which results in three images with the specified colors. They are: red, blue, yellow. The choice of color is due to the color of traffic signs. The result of filtering is a binary image in which the background is black and searched marks are white (pic. 27);



Pic. 27 Colors filtration by red, blue, yellow respectively

- further analysis is the geometric properties of objects that are found in the image. By these properties can be used to find exactly those objects that are required;
- the result of detection appears as an overlay on the original image boundaries of detected objects (pic. 28).

Pic. 28 Final result

Experimental results of this method allow us to assess the effectiveness of the method and some of its shortcomings. One of the advantages of this method is a relatively high speed. This result was achieved through the use of optimized functions in MATLAB. As a result, the average run-time processing and analysis of one frame does not exceed one second. Provided that the methods used can be further refined and simplified by removing a number of actions that do not affect the results of processing, processing speed may increase. In addition to the speed of operation we can also include its simplicity. The method does not use in its structure any complex methods and algorithms. A good fact is that the method shows a good detection rate. That is, thus proving its suitability of using. Test results are given in a table 3.1:

Table 3.1 Experimental results of the traffic signs detection method

Summary signs	Signs detection	Partly detection	Result, %
Day	130	10330	79
Night	87	59	67

In addition to the positive characteristics of the system, experimental studies revealed also some negative aspects, such as: the complexity of identifying the parameters of the limit color filter, delay in the morphological analysis, and quality of captured video stream.

3.5 Closing

According to the results of experimental studies, this method can be considered stable and effective in use. It allows us to solve the task of traffic signs detection without using complex and computations algorithms. Although the method has a number of shortcomings identified in the experimental studies, it is likely that a refinement of the method would eliminate these shortcomings, thereby the efficiency increase the method will be able to increase even more. Let us not also forget that existing methods of detection and recognition of road

signs in video stream are processing data in real time. Thus, we can say with full confidence that when the method that was developed and tested on a personal computer, works in real time, it can show good results, that are not inferior to existing methods.

Conclusion

In modern systems of vehicles safety, there are many technologies that could help us driving and notify the driver case of danger during car movement. Ability to constantly monitor traffic signs of restrictions and warnings on the road, leads to the fact that the driver is often distracted from vehicle control. Thus, it increases the likelihood of an accident. The solution, has been to actively develop systems to detect and recognize traffic signs and inform driver about them. Existing systems are complex computer systems, which are too costly to be implemented. To reduce the costs of implementation and introduction of systems of this type, it is necessary to develop methods of detection and recognition of road signs that might not reach the efficiency and speed of existing methods, but they have such advantages as: ease of implementation, low hardware requirements and ease of modification and introduction.

As a result of this work, we have developed a method of traffic sign detection, which is designed to solve several problems associated with reducing the cost of implementing this type of security systems for vehicles. In this method, there was implemented processes for gathering information through the capture of a video stream from an external device, the primary data processing to reduce noise and emerging algorithms for detecting the desired objects on the obtained data.

In the process of developing the method several existing processing techniques and image filtering have been successfully tested. As a result, a number of major algorithms on selected methods and included in the detection method, in order to improve its efficiency. The choice of the technology and the filters of the method was justified by theoretical analysis. Based on the information obtained in the analysis of existing methods and data that were obtained during the development of the detection method we have done as analysis and highlighted key features. In terms of these data and experiments in

the use of technologies in the developed method of discovery, we have arrived of findings on the feasibility of using those or other technologies and filters in the method to be developed.

During the development of the detection method of road signs in a video stream, we have made interim conclusions on the effectiveness of the implementation of this type of method, or possible implementation of algorithms and have selected one of the most effective methods. As a result, the final conclusions were made that the method has a number of distinctive features by which to achieve the targets and objectives set at the beginning of this work. The real effectiveness of this method can be judged only on the basis of experimental studies.

Software implementation of the method was conducted using the development environment MATLAB. The choice of the development environment was justified even before the development of the method, since this environment provides a set of standard packages that are designed for work with video and images, and also has a wide range of filters and data processing technologies. The method is a set of separate modules, which at the same time are separate functions. Thanks to a judicious construction of the hierarchy of functions, we have managed to achieve the best possible outcomes for data processing.

After the software implementation of the method of detection, it was necessary to check its working and evaluate the effectiveness of the method developed and the technology applied. The experimental results showed that the method is work quite effectively and has all the properties that were set for the method. But in addition to the positive side, there were also negative aspects of the method.

The results of this work show that the method of traffic signs detection is quite effective and can be applied in systems aimed at the recognition of road signs and displaying them to the driver. Although the method contains a

number of shortcomings that were discovered during the experimental analysis, these can probably be overcome by modifications to the method.

Appendix. Program code

Main.m:

```
%memory clear

clear all;

%% data obtaining by the video capture devices

str = VidCapInfo();

%features initialization

WebCam = VidSetup(str.Adaptor{2},1,str.Format{2,1}(10));

%% time of capture operations

times=30;

timeV=0;

%start capture

start(WebCam);

% cycle for the captured data from the memory

% of video capture devices and their processing

while(WebCam.FramesAcquired<times)

    % initialization data recorder for video capture devices

    trigger(WebCam);

    % obtain all data from the device capture

    img=getdata(WebCam);

    %% implementation of image processing
```

```
        t1=clock;
        % remain separate image obtained
        OrigImg = img(:,:,:,1);
        %preprocessing: normalization
        imgN = NightImProc(img(:,:,:,1));
        %preprocessing: Wiener filtration
        [imgN noise] = WienerF(imgN);
        % preprocessing: edge detection
        imgN = EdgeDt(imgN,noise);
        % obtain the coordinates for the three objects in the image
        Boxes = ObjDt(imgN);
        %result visualization
        Detect(OrigImg,Boxes);
        t2=clock;
        %one operation time calculation
        timeV(length(timeV))=etime(t2,t1);
end;
% stop capture
stop(WebCam);
clear memory
clear all;
```

VidCapInfo.m:

```
function [Out] = VidCapInfo()
% Function to print information on existing devices, video capture
%Adaptor - plug adaptors
% unction IMAQHWINFO provides a structure with a field InstalledAdaptors
% ist of all the adapters in this system, which has access to the application
imaqInfo = imaqhwinfo;
% obtain the number of available adapters, video capture
for VidCnt = 1:length(imaqInfo.InstalledAdaptors)
% obtain information on available video capture devices
VidInfo = imaqhwinfo(imaqInfo.InstalledAdaptors{VidCnt});
% counts the number of video capture devices
        for VidID = 1:length(VidInfo.DeviceIDs)
                % display information about video capture device
                Vid_info = imaqhwinfo(VidInfo.AdaptorName, VidID)
                % derivation of possible video formats
                Vid_info.SupportedFormats
                % preservation of possible adapters video capture
                Out.Adaptor = imaqInfo.InstalledAdaptors;
```

```
                Out.VidDevice{VidCnt,VidID}                                    =
                imaqhwinfo(VidInfo.AdaptorName, VidID);

                % preservation of possible formats video capture devices

                Out.Format{VidCnt,VidID} =  Vid_info.SupportedFormats;

        end

end

end
```

VidSetup.m:

```
function [VidCap] = VidSetup(AdaptorSetup,VidID,VideoFormat)

%AdaptorSetup - choice of video capture adapter (string)

%VidID - choice of video capture devices (number)

%VideoFormat - resolution video capture (string)

% Select video capture devices

VidCap = videoinput(AdaptorSetup,VidID,char(VideoFormat));

% choice of plants video capture devices

getselectedsource(VidCap)

% Setting the number of frames to record.

framesToLog = 2;

set(VidCap, 'FramesPerTrigger',framesToLog);

% setting the resumption of the video capture device

TriggerRpt = inf;
```

```
set(VidCap,'TriggerRepeat',TriggerRpt);

% configuration properties start video capture devices

triggerconfig(VidCap,'manual');

% operation is designed to improve the quality

% display of captured image

set(gcf,'doublebuffer','on');

% Access to the parameters of the capture device and set the frequency of frames.

% Parameter FrameRate refers to the particular characteristics of the device

% and can not be supported by other devices.

frameRate = 30;

src = getselectedsource(VidCap);

set(src, 'FrameRate', num2str(frameRate));

end
```

NightImProc.m:

```
function [Iout] = NightImProc(InputImg)

% function of converting the image obtained

% in low light conditions

%InputImg – input image

%Icor - corrected image
```

```
% transformation to a model RGB 256 colors
InputImg=double(InputImg)./255;
% if the brightness level is below 50% of the total scale
if(max(max(InputImg(:,:,2)))<0.5)
    % Extension of the brightness component image in the whole range from
    0 to 1
      Iout(:,:,1)=imadjust(InputImg(:,:,1),          [mean(min(InputImg(:,:,1)))
    mean(max(InputImg(:,:,1)))], [0.03 0.97], 1);
      Iout(:,:,2)=imadjust(InputImg(:,:,2),          [mean(min(InputImg(:,:,2)))
    mean(max(InputImg(:,:,2)))], [0.03 0.97], 1);
      Iout(:,:,3)=imadjust(InputImg(:,:,3),          [mean(min(InputImg(:,:,3)))
    mean(max(InputImg(:,:,3)))], [0.03 0.97], 1);
    %% Sharpen Image
    % initialization filter
    alpha = 0.5;
    h = fspecial('unsharp',alpha);
    % increase in the level of image sharpness
    Iout(:,:,1) = imfilter(Iout(:,:,1),h,'symmetric');
    Iout(:,:,2) = imfilter(Iout(:,:,2),h,'symmetric');
    Iout(:,:,3) = imfilter(Iout(:,:,3),h,'symmetric');
else
    Iout=InputImg;
```

```
end

end
```

WienerF.m:

```
function [Iout noise] = WienerF(InputImg)

% Adaptive Wiener filtering

%InputImg=im2double(InputImg);

% at approximately the level of noise as STD

sqrtV = std(InputImg(:,:,1),1);

sqrtV = mean(mean(sqrtV));

% Wiener filter, filtering operation for the three components RGB

[Iout(:,:,1) noise] = wiener2(InputImg(:,:,1),[3 3],sqrtV*0.1);

Iout(:,:,2) = wiener2(InputImg(:,:,2),[3 3],sqrtV*0.1);

Iout(:,:,3) = wiener2(InputImg(:,:,3),[3 3],sqrtV*0.1);

end
```

EdgeDt.m:

```
function [Iout] = EdgeDt(InputImg, k)

%edge detection

I=InputImg;

 %% by RGB
```

```
for s=1:3
        %components reading
        IoutA=InputImg(:,:,s);
        % average of the components
        Lser=mean(mean(IoutA));
        % correction on noise
        IoutA=Lser+(IoutA-Lser)*k*70;
        %changing
        Iout(:,:,s)=IoutA;
end
end
```

ObjDt.m:

```
function [Boxes] = ObjDt(InputImg)
% function determining whether the unknown objects by their geometric shape
% count the number of found objects
Index=1;
% structure to store the coordinates of the outlines of the detected objects
Boxes=struct('Box',[]);
for s=1:3
        %color filtration
```

```
img = double(ColorDt(InputImg,s));

% determine the number of objects in the image

[imgN num] = bwlabel(img,4);

% obtain the properties of detected objects

f = imfeature(imgN,'BoundingBox','Centroid','Extent','Eccentricity',4);

%square

for i=1:num

    if(f(i).Extent>=0.78) & (f(i).Extent<0.95) & (f(i).Eccentricity<0.9)
    & (f(i).Eccentricity>=0.2)

        Boxes(Index).Box=f(i).BoundingBox;

        Index=Index+1;

    end

end

%circle

for i=1:num

    if(f(i).Extent<0.78) & (f(i).Extent>=0.65) & (f(i).Eccentricity<0.8)
    & (f(i).Eccentricity>0.3)

        Boxes(Index).Box=f(i).BoundingBox;

        Index=Index+1;

    end

end

%triangle
```

```
        for i=1:num
            if(f(i).Extent>0.3) & (f(i).Extent<0.6) & (f(i).Eccentricity<0.8) &
            (f(i).Eccentricity>0.2)
                Boxes(Index).Box=f(i).BoundingBox;
                Index=Index+1;
            end
        end
end
end
```

ColorDt.m:

```
function [Img] = ColorDt(InputImg,ColorInd)
%color filtration
%receipt of the imageя
%InputImg=double(InputImg)./255;
[N M s]=size(InputImg);
% morphological image analysis
se=strel('disk',5);%diamond ball square disk
inImR=imerode(InputImg, se);
InputImg1=imreconstruct(inImR, InputImg);
InputImg=imdilate(InputImg1, se);
```

```
inIm=imreconstruct(imcomplement(InputImg), imcomplement(InputImg1));
InputImg=imcomplement(inIm);
%color filtration
if(ColorInd==1)
figure,imshow(InputImg);
ImgR = InputImg;
for i=1:N
    for j=1:M
    %red
        if (ImgR(i,j,1)>=0.3) & (ImgR(i,j,2)<=0.45) & (ImgR(i,j,3)<=0.45)
            ImgR(i,j,1)=1;
            ImgR(i,j,2)=1;
            ImgR(i,j,3)=1;
        else
            ImgR(i,j,1)=0;
            ImgR(i,j,2)=0;
            ImgR(i,j,3)=0;
        end;
    end;
end;
Img = DelNoise(ImgR);
```

```
figure,imshow(Img)
end;
if(ColorInd==2)
ImgB = InputImg;
for i=1:N
    for j=1:M
    %blue
        if (ImgB(i,j,1)<=0.45)& (ImgB(i,j,2)<=0.45) & (ImgB(i,j,3)>=0.2)
            ImgB(i,j,1)=1;
            ImgB(i,j,2)=1;
            ImgB(i,j,3)=1;

        else
            ImgB(i,j,1)=0;
            ImgB(i,j,2)=0;
            ImgB(i,j,3)=0;
        end;
    end;
end;
Img = DelNoise(ImgB);
figure,imshow(Img)
```

```
end
if(ColorInd==3)
ImgY = InputImg;
for i=1:N
        for j=1:M
                %yellow
                if (ImgY(i,j,1)>=0.6)& (ImgY(i,j,2)>=0.6) & (ImgY(i,j,3)<=0.3)
                        ImgY(i,j,1)=1;
                        ImgY(i,j,2)=1;
                        ImgY(i,j,3)=1;
                else
                        ImgY(i,j,1)=0;
                        ImgY(i,j,2)=0;
                        ImgY(i,j,3)=0;
                end;
        end;
end;
Img = DelNoise(ImgY);
figure,imshow(Img)
end
 %figure,imshow(ImgR);
```

```
%figure,imshow(ImgB);

%figure,imshow(ImgY);

end
```

DelNoise.m:

```
function [Iout] = DelNoise(InputImg)

% reducing the availability of foreign objects in the image

 % conversion in binary form

InputImg=im2bw(InputImg);

% filling the cavities of objects

InputImg=imfill(InputImg,'holes');

% increasing the thickness of lines

se90=strel('line', 3, 90);

se0=strel('line', 3, 0);

Iout=~(imdilate(~InputImg, [se90 se0]));

% Remove objects from images, measuring less than 1000 pixels

Iout=bwareaopen(Iout,1000,4);

end
```

Detect.m:

```
function [] = Detect(InputImg,Boxes)
```

```
% allocation of the detected objects in the image

hold off;

imshow(InputImg);

hold on;

if(length(Boxes(1).Box)>0)

        for i=1:length(Boxes)

                %left vert

                line([Boxes(i).Box(1)                Boxes(i).Box(1)],[Boxes(i).Box(2)
                Boxes(i).Box(2)+Boxes(i).Box(4)] );

                %up horz

                line([Boxes(i).Box(1)
                Boxes(i).Box(1)+Boxes(i).Box(3)],[Boxes(i).Box(2)
                Boxes(i).Box(2)] );

                %right vert

                line([Boxes(i).Box(1)+Boxes(i).Box(3)
                Boxes(i).Box(1)+Boxes(i).Box(3)],[Boxes(i).Box(2)
                Boxes(i).Box(2)+Boxes(i).Box(4)] );

                %down horz

                line([Boxes(i).Box(1)
                Boxes(i).Box(1)+Boxes(i).Box(3)],[Boxes(i).Box(2)+Boxes(i).Box
                (4) Boxes(i).Box(2)+Boxes(i).Box(4)] );

        end;

end end
```

References

1. J. Hatzidimos, 2004. Automatic traffic recognition in digital images. Conference on Theory and Applications of Mathematics and Informatics.
2. H. Fleyeh, M. Dougherty, 2006. Road And Traffic Sign Detection And Recognition. Proceedings of the 16th Mini - EURO Conference and 10th Meeting of EWGT. pages 644-653.
3. Shojania H., 2007. Real-time Traffic Sign Detection. Private publication.
4. Moutarde F., Bargeton A., 2005. Modular traffic signs recognition applied to on-vechile real-time visual detection of USA and EU speed limit signs. IEEE Intelligent Vehicles Symposium. Las Vegas.
5. Lauffenburger J., Bradai B., 2008. Navigation and Speed Signs Recognition Fusion for Enhanced Vehicle Location. Proceedings of the 17th World Congress.
6. Moutarde F. Bargeton A., 2007. Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular Traffic Signs Recognition system. IEEE Intelligent Vehicles Symposium Proc., pages 1122-1126, Istanbul.
7. David G. Lowe, 2004. Distinctive image features from scale-invariant keypoints. Journal of Computer Vision, № 60, pages 91-110.
8. Ihara A., Fujiyoshi H., 2007. Improved Matching Accuracy in Traffic Sign Recognition by using SIFT Features. Publication of Denso Corporation. pages 824-831.
9. Perona P., Zisserman A., 2007. Object Class Recognition by Unsupervised Scale-Invariant Learning. IEEE Intelligent Vehicles Symposium Proc.
10. Chen M., Gao T., 2008. Detection and Recognition of Alert Traffic Signs. Publication of Standford University.
11. Hi-Tech Security company [online document]. [Accessed 2010], Available at http://www.hitsec.ru/articles.htm

12. Matlab Image Processing Toolbox Functions: imwrite [online document]. [Accessed 2010], Available at http://www.mathworks.com/access/helpdesk/help/techdoc/ref/imwrite.html
13. Bracewell R.N., 1978. The Fourier Transform and its Applications. McGraw-Hill, N.Y.
14. Webcam features [online document]. [Accessed 2010], ArmoSystems, Available at http://www.armosystems.ru/support/faq/2.ahtm
15. How to choose webcam [online document]. [Accessed 2009], Available at http://www.003.ru/articles/read/608.html#shag2
16. Asmakov S. 2001. USB 2.0 [online document]. [Accessed 2010], Computer Press, Available at http://www.compress.ru/article.aspx?id=12032&iid=464
17. YUV&RGB [online document]. [Accessed 2008], Available at http://provegas.ru/forum/showthread.php?t=360
18. Color depth [online document]. [Accessed 2007], Available at http://en.wikipedia.org/wiki/Color_depth
19. Vatolin D., Ratushnyak A., 2002. Compression methods. Dialog, MIFI, Moskow.
20. Thompson W.B., Shirley P., 2002. A Spatial Post-Processing Algorithm for Images of Night Scenes.
21. Zhuravel I., 2009. Methods of the image normalization [online document]. [Accessed May 2010], Available at http://matlab.exponenta.ru/imageprocess/book2/52.php
22. Polyakov A., Brusencev V., 2003. Alghoritms and methods of computer graphics on Visual C++. BVH-Petersburg, Russia.
23. Sahtarin B., 2008. Wiener and Calman filters. Grif Russian Univers. Gelios ARB.
24. Canny J.E., 1986. A computational approach to edge detection. IEEE Trans Pattern Analysis and Machine Intelligence. No. 8, pages 679 — 698.
25. Gonsales R., Woods P., Ediss E., 2006. Digital image processing on MATLAB. Digital world, Technosphere, chapter 10.

26. Zhuravel I., 2009. K-means segmentation [online document]. [Accessed May 2010], Available at http://matlab.exponenta.ru/imageprocess/book2/31.php
27. Zhuravel I., 2010. Ыegmentation method of controlled watershed [online document]. [Accessed May 2010], Available at http://matlab.exponenta.ru/imageprocess/book2/48.php
28. Kaas M., Witkin A., Terzopoulos D., 1987. Snakes Active Contour Models. Int. Journal of Computer Vision, vol.1, no.4, pages 312-331.
29. Zhuravel I., 2010. Texture segmentation [online document]. [Accessed May 2010], Available at http://matlab.exponenta.ru/imageprocess/book2/55.php
30. Zhuravel I., 2009. Recognition based on the features [online document]. [Accessed May 2010], Available at http://matlab.exponenta.ru/imageprocess/book2/58.php
31. Arias-Castro E., Donoho D.L., 2006. Does median filtering truly preserve edges better than linear filtering? Annals of Statistics, vol. 37, no.3, pages 1172–2009.
32. Matlab Image Processing Toolbox Functions: getdata [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/daq/getdata.html
33. Matlab Image Processing Toolbox Functions: imaqhwinfo [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/imaq/imaqhwinfo.html
34. Matlab Image Processing Toolbox Functions: videoinput [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/imaq/videoinput.html
35. Matlab Image Processing Toolbox Functions: imadjust [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/imadjust.html

36. Matlab Image Processing Toolbox Functions: fspecial [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/fspecial.html
37. Matlab Image Processing Toolbox Functions: imfilter [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/imfilter.html
38. Matlab Image Processing Toolbox Functions: wiener2 [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/wiener2.html
39. Matlab Image Processing Toolbox Functions: imerode [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/imerode.html
40. Matlab Image Processing Toolbox Functions: imreconstruct [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/imreconstruct.html
41. Matlab Image Processing Toolbox Functions: strel [online document]. [Accessed May 2010], Documentation, Available at http://www.mathworks.com/access/helpdesk/help/toolbox/images/strel.html
42. Matlab Image Processing Toolbox Functions: imfeature [online document]. [Accessed May 2010], Documentation, Available at http://www.math.carleton.ca/old/help/matlab/MathWorks_R13Doc/toolbox/images/imfeature.html
43. Beits P., Mac-Donnell M., 1989. Image restore and reconstruction. page 336.