Image Processing in Road Traffic Analysis*

E. Atkočiūnas[1], **R. Blake**[2], **A. Juozapavičius**[1], **M. Kazimianec**[1]

[1]Faculty of Mathematics and Informatics, Vilnius University, Lithuania
erikas.atkociunas@gmail.com; algimantas.juozapavicius@maf.vu.lt; michailas@satela.lt

[2]Department of Computer and Information Sciences, NTNU, Norway
blake@rover.idi.ntnu.no

Received: 27.08.2005
Accepted: 18.11.2005

Abstract. The article presents an application of computer vision methods to traffic flow monitoring and road traffic analysis. The application is utilizing image-processing and pattern recognition methods designed and modified to the needs and constrains of road traffic analysis. These methods combined together gives functional capabilities of the system to monitor the road, to initiate automated vehicle tracking, to measure the speed, and to recognize number plates of a car. Software developed was applied in and approved with video monitoring system, based on standard CCTV cameras connected to wide area network computers.

Keywords: computer vision, flow monitoring, traffic analysis, image processing, vehicle tracking, speed measurement, number plate recognition, motion detection, contour extraction, contour labeling, filtration, threshold, mask, LPR.

1 Introduction

Traffic flow monitoring and traffic analysis based on computer vision techniques, and especially traffic analysis and monitoring in a real-time mode raise precious and complicated demands to computer algorithms and technological solutions. Most convincing applications are in vehicle tracking, and the crucial issue is initiating a track automatically. Traffic analysis then leads to reports of speed

*This research is partially supported by VMSF – Lithuanian Science and Studies Foundation, Reistration No. B-03027.

violations, traffic congestions, accidents, or illegal behaviour of road users. Various approaches to these tasks were suggested by many scientists and researchers [1–3].

The approach in this article focuses on methods of image processing, pattern recognition and computer vision algorithms to be applied to road traffic analysis and monitoring. One of the main aspects was to modify these algorithms to fit to real-time road monitoring processes, and as a consequence the prototype of system for traffic analysis was developed. Technically this system is based on stationary video cameras as well as computers connected to wide area network.

Capabilities of the system include vehicle tracking, vehicle speed measurement (without use of traditional sensors), and recognition of license plate numbers of moving vehicles, lane jam detection. Additional features of the system are object/data searching and archiving, statistical analysis. Image processing tasks utilized in the system are image filtering, correction and segmentation, object modeling, tracking and identification, morphological, geometrical and statistical methods. Technical tasks used are motion shooting, video sequence transmitting, frame extraction.

2 Problem description

Image processing and object/pattern recognition of moving objects, chosen for the system, lead to complex mathematical, algorithmic and programming problems. Many articles (see for instance [4]) have considered particular questions related: scene modeling, object geometry accounting, image contours processing. There is a lack of information on methods and algorithms used in digital monitoring technology, perhaps for commercial reasons.

The problem of road monitoring as it is chosen in our research is presented as a sequence of independent processing steps intended to solve tasks logically connected to each other.

These steps are in hand of the following order of algorithmic processing: video stream input to computer (personal computer or specialized one), its conversion to a sequence of single frames, lanes masking, background removal, noise and blobs filtering, object contours extraction, linking and labeling, contour parameters estimation, moving vehicle tracking, velocity calculation, lane congestion estimation, number plate recognition.

Such a sequence of steps is determined by the order of logical stages. For the first of all initial data have to be given in the form of video sequence, and then processed to locate observed vehicle in each frame. The imaging procedures like segmentation, filtering and edge detection ones are arranged and utilized, that allocate vehicle contour within frame observation zone. The next stage is to find and mark a conditional center of vehicle presented by its contour area, in order to calculate speed of an object, and to track it within frames. Third stage is to label contours, which helps us to mark and calculate a number of moving objects in the observation zone, and thus estimate lane congestion. Next stage gives a possibility to detect vehicle type, by using matching criteria for comparison of vehicle contours segmented to typical outlines of cars, trucks, pickups and buses. The last stage consists of capture and recognition of vehicle number plates.

There may be two types of data used in the system, related to the location of stationary camera: motion scenes may be filmed with a view from above to the road surface, or motion scenes may be filmed from road level, under the fixed angle to vehicle motion. In the first case data are suitable for vehicle tracking and speed measurements, and in the second case – for number plate recognition. Notice that in the system there are no sensors used or electromagnetic loops installed in the roadbed to detect moving vehicles.

3 Methods and algorithms

There are two different technologies suitable for road traffic processing:

- moving object tracking and speed measurement,
- automatic number plate registration and recognition.

To analyze vehicle tracking in video sequence two methods will be compared:

- object definition based on object contour extraction,
- object definition based on motion detection.

3.1 Vehicle tracking based on contour extraction

Corresponding algorithms are based on frame segmentation, object contour finding and labeling. They can be divided into six steps, each having specific processing algorithm.

Lane masking. This method is intended to separate the part of the road where vehicles are moving in one direction (Fig. 1). This action is essential because of it let's to simplify an information processing extracted from more than one frame. Masking algorithm is given by formula

$$N(p) = M(p) \times V(p),$$

where $M(p)$ is an image point value in primary frame, $N(p)$ is a new image point in the output image (Fig. 2), $V(p)$ is mask value for point p: $V(p) = 0$ if corresponding pixel is eliminated, otherwise $V(p) = 1$. Masking is applied to each RGB color separately.

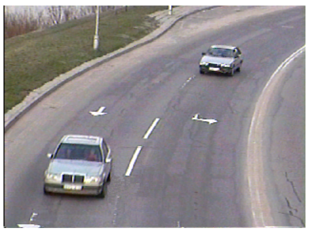


Fig. 1. Before masking.

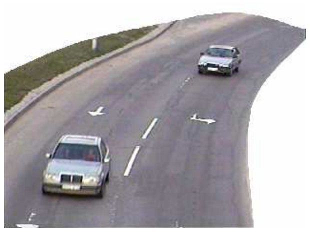


Fig. 2. After masking.

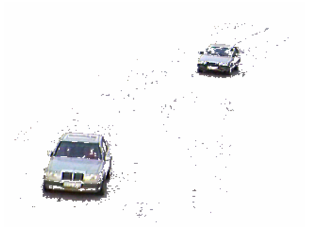


Fig. 3. After subtraction.

It is convenient to constrain lane mask by using graphical editor especially created for system configuration. This feature connects sharp points given by user into persistent curve.

Background elimination. This algorithm removes all stationary objects from lane observation zone leaving only vehicles and some details, which are changing from frame to frame. The latter are influenced usually by image noise and some background factors: swinging trees, moving grass, light shadows, clouds, rain, snow etc. In real conditions time, season, weather and some others factors must be also taken into account.

Background $B(p)$ is calculated as an average of each RGB values for the same image point p in the selected background frames:

$$B(p) = \sum_k \frac{I_B(k,p)}{n},$$

where $I_B(k,p)$ is pixel color value for point p in frame k. Background removal from traffic scene image $I(k,p)$ generates a color RGB image

$$D(k,p) = \big\{I(k,p) - B(p)\big\},$$

as an Euclidean distance $\{.\}$ between $I(k,p)$ and $B(p)$. Fig. 3 illustrates this result.

Noise and blobs filtration. Image, as presented in Fig. 3 has a lot of speckles caused by noise, which could be removed only by means of filtration. An effective method is threshold filtration just after background elimination. There are two main variations of this method:

- fixed threshold – when it is determined empirically from the test sequence,
- histogram-derived threshold – when the latter is selected from the brightness histogram of the segmented region.

The first algorithm is more adaptive one and is generating an additional image:

$$M(k,p) = D(k,p) \text{ if } D(k,p) > threshold \text{ and } M(k,p) = 0 \text{ for the rest.}$$

Next step removes isolated points and little blobs that remain after threshold filtration. The known idea is simple enough. If an image point p belongs to object then at least one of its eight adjacent points is a part of this object. If its adjacent points do not belong to object then so the point p does. This method was modified for the method under consideration by removing all points inside little $m \times n$ sliding rectangular window in case when its one pixel width contour has white color. In binary image it indicates the absence of points. Eight adjacent points for a pixel were considered here, instead of four adjacent points (as it is defined in many computer graphics algorithms). Such situation is caused by requirements for resolution, suitable for stable contour detection.

Remaining noise can be effectively reduced by median filtration. Let's suppose there is a $m \times n$ mask applied to the area that surrounds original point $M(k,p)$. Forming up a sequence of $m \times n$ elements in the progressive order

$$I_1 < I_2 < \ldots < I_k, \quad \text{where} \quad k = m \times n,$$

an element I_J may be find from mask area which is the nearest to the center of this sequence. This pixel substitutes $M(p,k)$. Fig. 4 and Fig. 5 illustrate these types of filtration. These operations are the last in a frame preprocessing stage.

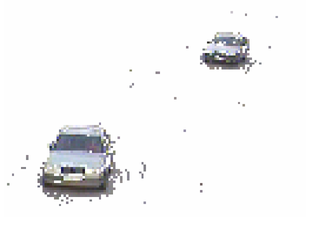


Fig. 4. Blobs filtering.

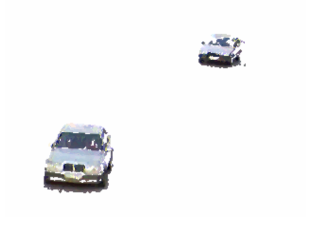


Fig. 5. After 3×3 median filtering.

Contour extraction. Now there is a need to locate a vehicle in frame observation zone. It can be done by edge detection algorithm based on the calculation of image derivatives. They allow examining whether edge passes through a given pixel – steep gradients are evidence of an edge, slow changes suggest the contrary.

First derivative calculation is intended to detect the rate of intensity changes in frame images. Second derivative is used to determine the location of the maximum rate. To reduce derivative leaps within and near contour area initial image is smoothing.

It is important to note that there is no universally detection technique that works equally well for all images. Therefore we have compared most widespread detection algorithms based on Prewitt, Sobel and Laplace methods [4, 5] that are applied usually after median filtration.

All methods mentioned are referring to mask (kernel) operations that provide the following results: gradient S_x in X direction, gradient S_y in Y direction and module M of gradient:

$$S_X = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6),$$
$$S_Y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4),$$

where $M = (S_x^2 + S_y^2)^{1/2}$, $c = 2$ and a_i – pixels covered by 3×3 mask area. The designation $a_0, a_1, \ldots, a_7$ refers to the clockwise motion beginning from upper left corner of the mask. If $M > threshold$ then $M(k,p) = 1$ else $M(k,p) = 0$.

The kernel is applied almost to all the pixels of the image except edge area: resulting picture is reduced in size by two pixels in each direction because of border effects. Graphical presentation of Sobel edge operator kernels is given at Fig. 6, where symbol “$\leftarrow$” means local derivative calculated as an intensity difference between two adjacent points (the distance between them is equal to 1).

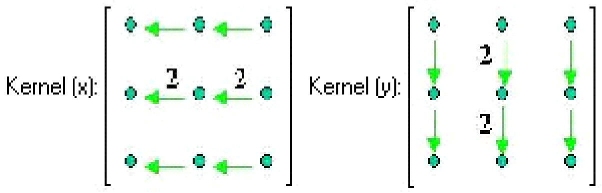


Fig. 6. S_x and S_y kernels.

Prewitt filter kernel is very similar to Sobel. The only difference is weighing coefficient c that in case of Prewitt filter is equal 1. Fig. 7 and Fig. 8 illustrate Prewitt and Sobel filtration results applied to car edge detection. The advantage of these methods is that the computation is a fast one. If initial images are well contrasted and noiseless they are perspective for edge extraction.

As mentioned above, gradient is considered large when its magnitude is larger than a threshold. Another way is to assume that gradient is large whenever second order intensity gradients have a zero crossing. Two mostly widespread Laplace algorithms were considered: Zero crossing edge detection (Fig. 9) that compares gradient magnitude according to zero crossing operation and Laplacian of Gaussian (LoG) detection with additional derivative smoothing performed by Gauss convolution (Fig. 10):

$$I(i,j) = \sum_{l,k} S(l,k) \times G(i-l, j-k), \quad \text{where} \quad G(i,j) = \frac{\pi A}{2} \times \exp\left(-\frac{i^2+j^2}{2A^2}\right),$$

where A is a width of Gauss function G. As we see Fig. 7 has best quality. Therefore Prewitt algorithm was selected as basic for edge detection.

Contour linking. Next processing step is contour linking connecting separated edge parts of the original object into one closed contour. For convenience suppose that RGB image is transformed to binary one

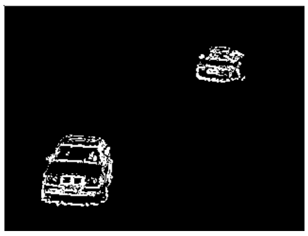


Fig. 7. Prewitt edge detection.

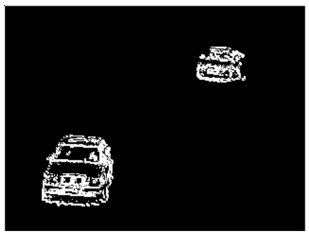


Fig. 8. Sobel edge detection.

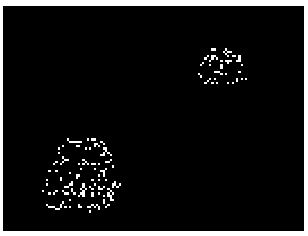


Fig. 9. Laplacian zero crossing.

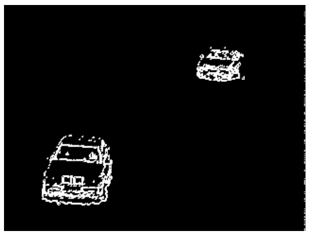


Fig. 10. LoG Edge Detection.

In general edge-linking methods can be classified into two categories. Local edge linkers group points by considering each point's relationship to any neighboring edge points. Global edge linkers consider all edge points at the same time according to some similarity constraint, for example, to edge equation.

Local edge linkers start at some arbitrary edge point and consider adjacent points (Fig. 11). If points satisfy given a criterion, which refers to gradient magnitude and direction, they are added to the current edge set. We simplified linking referring to the contour points that are no more than two pixels distant away from the point declared earlier as contour member (Fig. 12).

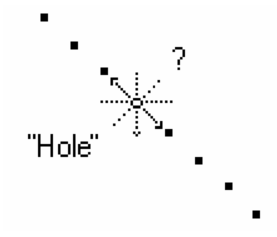


Fig. 11. Edge linking.

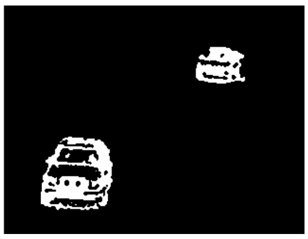


Fig. 12. Hole elimination.

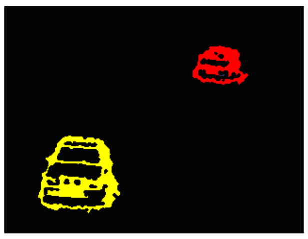


Fig. 13. Contour labeling.

The global edge linking methods are more complicated. They are used to get more correct results and let to avoid the merging of different contours into one object, and their weakness is a sophisticated calculation.

Contour labeling. This method is used to mark and calculate vehicles within frame. To solve a problem we modified the region growing method. The latter is applied to object contour of one pixel width, while the region growing method usually is applied to object area. The result of the proposed algorithm is shown at Fig. 13. Calculations start by marking first selected contour point by certain color. Then the primary color of this point is compared with the color of the adjacent point. If both colors are identical adjacent point is also labeling by new color. Curve is lengthened pixel to pixel by adding the neighboring points.

Vehicle tracking. To track vehicle in the video sequence we must mark its image in someway. One of the ways is to mark object geometric center that is calculated as

$$x_c = \frac{\sum x_j}{n}, \quad y_c = \frac{\sum y_j}{n},$$

where x_c and y_c are vehicle center coordinates and x_j, y_j – are coordinates of one of n image points from the area limited by vehicle external contour.

To analyze tracking assume that vehicles do not outride together within video camera watching zone and that the displacement of image center of the observed vehicle in two neighboring frames is less than the distance between its and another vehicle centers in the same or neighboring frames.

To track vehicle we must calculate all distances d_k between vehicle image in frame n and all vehicle images in frame $n+1$ using coordinates (x_k, y_k) of their centers:

$$d_k = \left[(k_k - x_c)^2 + (y_k - y_c)^2\right]^{\frac{1}{2}}.$$

Calculating $d = \min(d_k)$ we find tracking vehicle in frame $n+1$. Applying this method to all frames we monitor vehicle in video sequence.

3.2 Vehicle tracking based on motion detection

This method is based on the analysis of sensitivity zones of video camera matrices. The principle is close to video surveillance technology. Every frame is segmented into $(N \times M)/(n \times m)$ mesh where $(n \times m) \ll (N \times M)$, $N \times M$ – number of pixels in the frame, $n \times m$ – number of pixels in the detection eyehole. It is intended that eyehole detects motion if average of its pixel intensities exceeds given threshold.

All motion-detected pinholes are analyzed and assorted according to the dependence to the compact area criteria.It lets the algorithm to fix moving vehicle position in each frame and as a result in the video sequence on the whole sequence. Dependency may be calculated using stochastic adjacency criteria and previous frame information.

Fig. 14 demonstrates the result of the application of this method to frames with different resolutions: $N \times M = 384 \times 288$, $n \times m = 4 \times 4$, id est when the number of eyeholes is 96×72. Fig. 14a illustrates motion zone and moving object center detection; Fig. 14b and 14c are referring to next processing steps: speed estimation and jam detection.

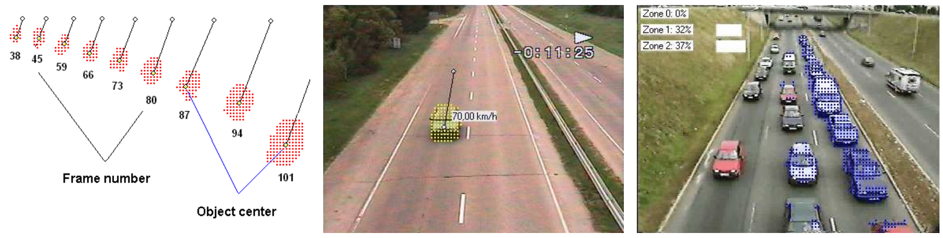


Fig. 14. Object tracking (a), speed estimation (b) and jam detection (c) by motion detection method.

Described method is enough effective for object tracking but it is not accurate for speed measurement. We advise to apply it to qualitative transport flow analyses.

Speed measurement. Vehicle speed v was estimated using below described method.

First we made road experiment. Motion was filmed at the same road length for some car drives with known speeds $v_1 < v_2 < v_3 < \ldots < v_n$.

Then every video sequence was processed to get passed distance S dependence from frame number (Fig. 15) calculated as a sum of car displacements fixed in n previous sequence frames (Fig. 16):

$$S = a_1 + a_2 + a_3 + \ldots + a_n,$$

where a_i is a distance between car image geometrical centers in the i and $i-1$ frame:

$$a_i = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}.$$

Here x_k ir y_k, $k = i-1, i$ are horizontal and vertical coordinates of car image center.

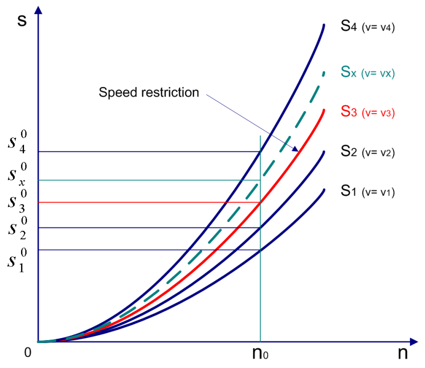


Fig. 15. Car video way as function of frame number.

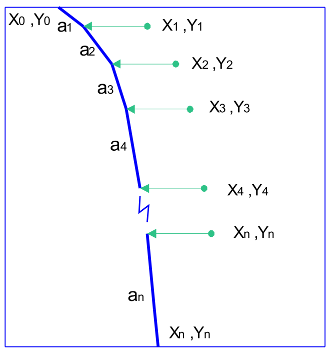


Fig. 16. Video way approximation.

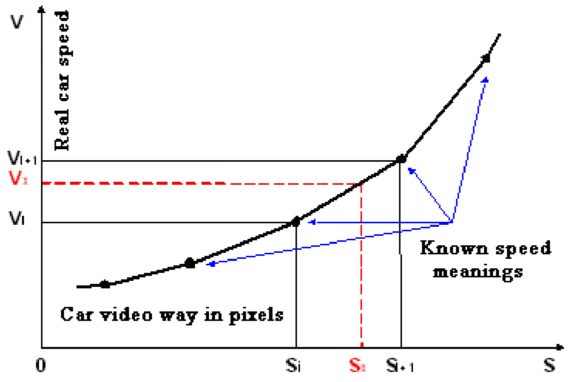


Fig. 17. Speed function approximation.

Experimental curves $S_i = s_i(n)$ allow to estimate a speed of any vehicle that moves within calibrated road segment. Estimation is based on function $v = v(s)$ for given frame number $n = n_0$.

$$V_x = A_i * S_x + B_i, \quad A_i = (V_{i+1} - V_i)/(S_{i+1} - S_i),$$
$$B_i = (V_i * S_{i+1} - V_{i+1} * S_i)/(S_{i+1} - S_i).$$

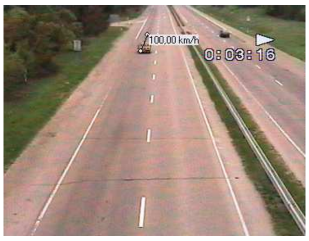
Speed meaning may be corrected by v_x average for different n_0.

Shown picture allows fixing speed violation. If analyzed car function is above red curve (Fig. 15) it means speed overrun.

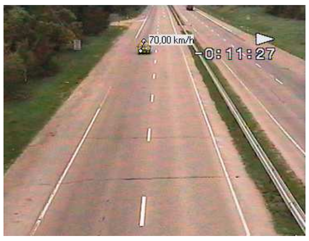
To get calibration function $v = v(s)$ corresponding to video sequences, 50 frames of each sequence were processed. To demonstrate real speed measurement test drives with 90 km per hour were made. Function $v = v(s)$ was calculated for frame numbers 26, 41 and 51. The estimated average of according three speed meanings is $92,15$ km/hour with measuring error 2,39 %.

Moving car images

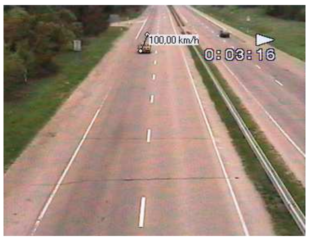
Frame No. 1:
60 km per hour



Frame No. 1:
70 km per hour

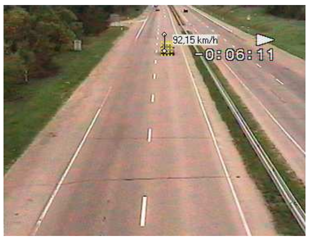


Frame No. 1:
100 km per hour

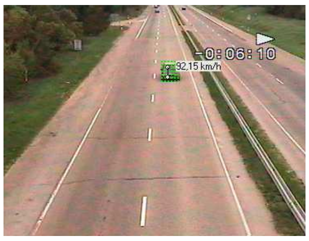


Speed estimation

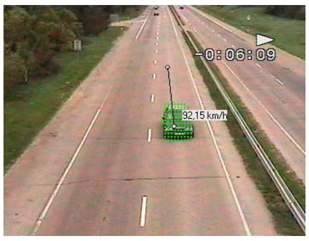
Frame No. 1:
$92,15$ km per hour



Frame No. 26:
$92,15$ km per hour



Frame No. 51:
$92,15$ km per hour

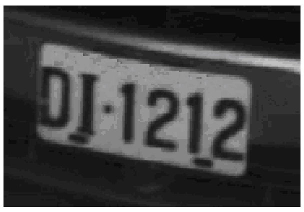


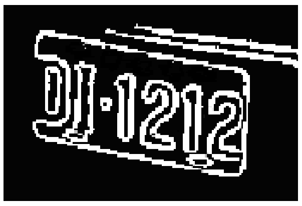
3.3 Number plate registration and recognition

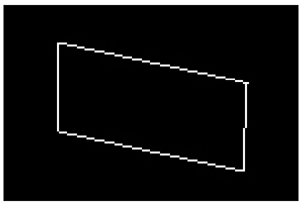
Now the automatic number plate registration and recognition technology is to be analysed. This technology has following peculiarities:

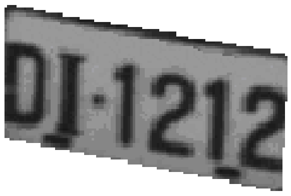
- video camera is oriented to fix front or rear (or both) vehicle license plate;
- when car moves through the observational zone recognition program localize number plate and tracks it while car is in this zone.
- symbol recognition module selects frame of the best quality and identifies it as symbol combination.
- recognized number plate, data, time and car image are writing to system database.

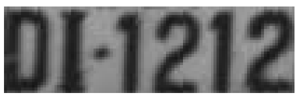
Processing problem is related to symbol extraction from number plate image and further symbol recognition. The algorithm used is described as follows:

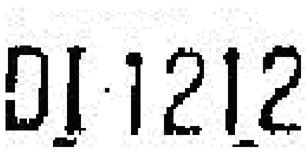

1. Number plate image

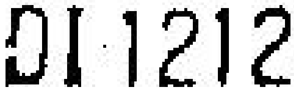

2. Contour extraction

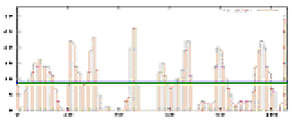

3. Processing area

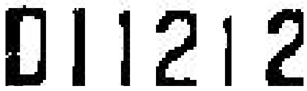

4. Numer plate cut up


5. Number plate deflection


6. Background removal


7. Noise filtration


8. Image histogram


9. Symbol separation

For the first of all, number plate edges and their intersection coordinates were found by applying Hough transform. After image filtration and turn number plate symbol intensity histogram was calculated. Then it was processed for given threshold to extract image of each number plate symbol. At the last stage neural network processing technology was applied to recognize them as text symbols.

Number plate recognition is a complex mathematical and algorithmic problem whose solution depends on many factors such as image quality and format, day time, illumination, weather conditions etc. It is clear that not all of them could be taken into consideration. Below we show examples of normal and hardly processing frames.

Normal image

Confusing format

Confusing background

Bad frame

Bad angle

Weak reflection

Bad illumination

Turned on light

Blocked number

Foggy image

Crooked plate

Extraneous records

In practice processing efficiency can be improved by organizational and technical efforts. Thus authorities can require drivers to vouch that license plates be clear and not twisted. From the other side processing accuracy may be improved applying optimal orientation of video camera, necessary light filters and good watching zone illumination.

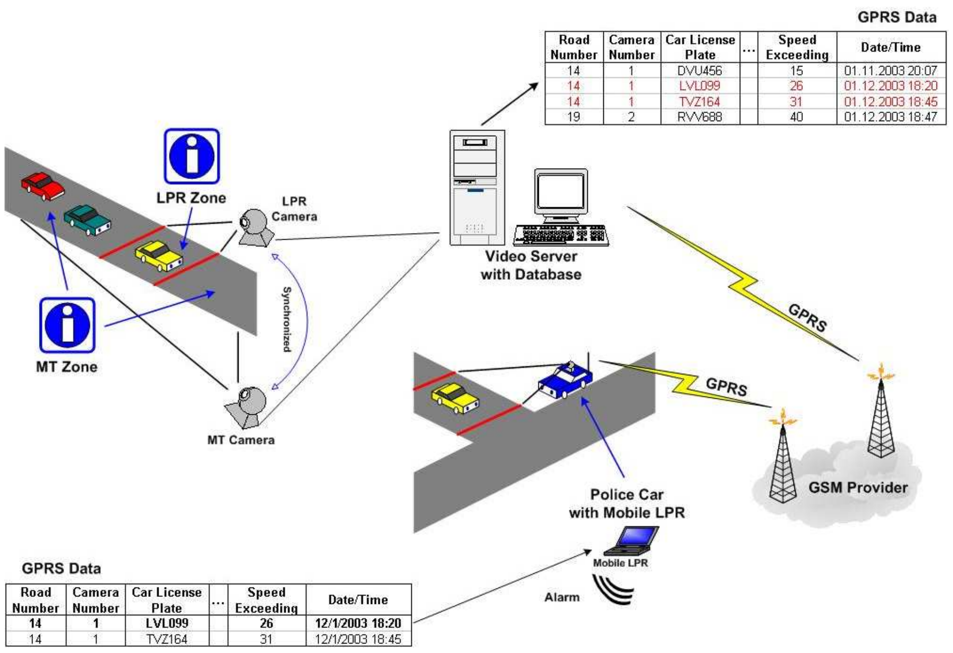
4 The system

Above described principles were implemented in the complex video system consisting of three main parts: vehicle tracking, number plate recognition, and monitoring center with video server and central database subsystem (Fig. 18) communicating over computer network.

Vehicle tracking subsystem is intended to watch transport motion, detect jams, and determine speed violation. If speed exceeds permissible one subsystem begins to track vehicle turning on LPR subsystem that files and transmits number plate data, violation time and date to central database and mobile LPR groups. The latter stop violator, verify data given from him and central database, prescribe penalty. As it is shown information interchange between LPR system installed in mobile group's notebook and monitoring center is based on data transmission over GPRS network. In case of jam tracking system informs monitoring center and transmits to it traffic video clip.

User interface of vehicle tracking and speed measurement subsystem is shown at Fig. 19, LPR subsystem interface – at Fig. 20.

All system applications are written in C++. Part of them is realized on the base of "Mega Frame" and "Carmen program libraries".



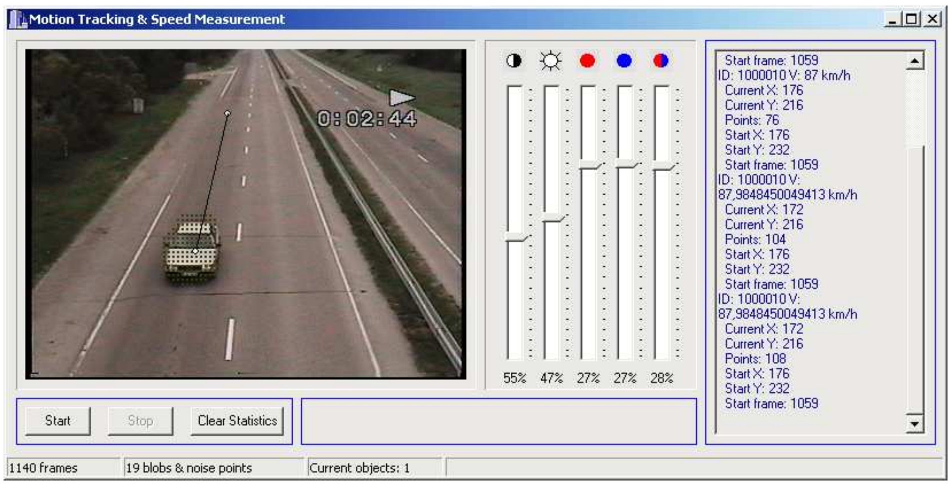
GPRS Data

Road Number	Camera Number	Car License Plate	...	Speed Exceeding	Date/Time
14	1	DVU456		15	01.11.2003 20:07
14	1	LVL099		26	01.12.2003 18:20
14	1	TVZ164		31	01.12.2003 18:45
19	2	RVV688		40	01.12.2003 18:47

GPRS Data

Road Number	Camera Number	Car License Plate	...	Speed Exceeding	Date/Time
14	**1**	**LVL099**		**26**	**12/1/2003 18:20**
14	1	TVZ164		31	12/1/2003 18:45

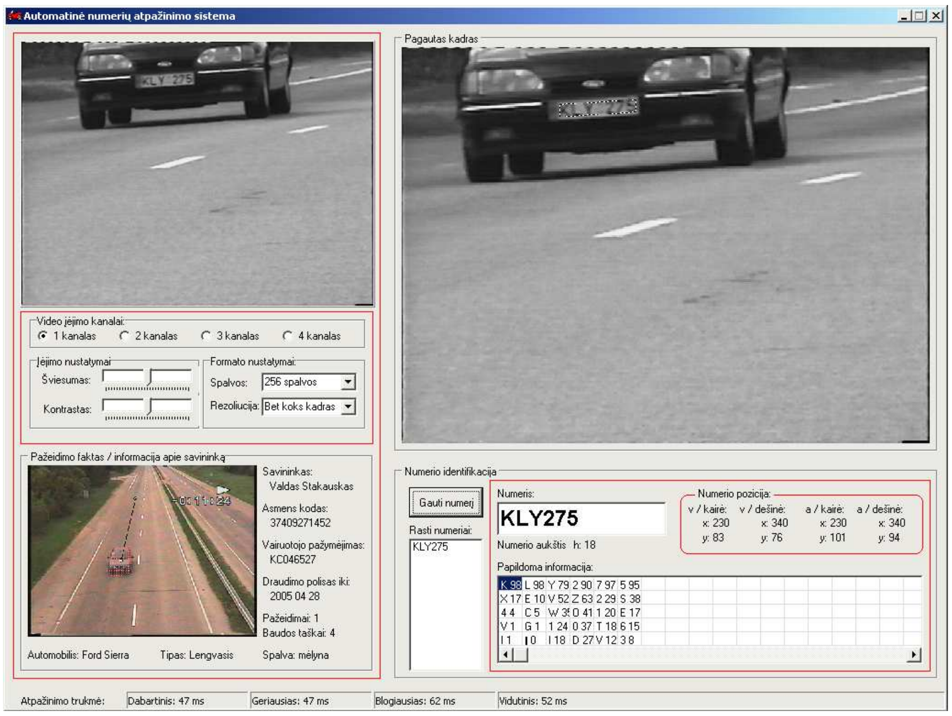
Fig. 18. Monitoring system. MT – Motion tracking, LPR – License plate recognition, GPRS – General packet radio service.



```
  Start frame: 1059
ID: 1000010 V: 87 km/h
  Current X: 176
  Current Y: 216
  Points: 76
  Start X: 176
  Start Y: 232
  Start frame: 1059
ID: 1000010 V:
87,9848450049413 km/h
  Current X: 172
  Current Y: 216
  Points: 104
  Start X: 176
  Start Y: 232
  Start frame: 1059
ID: 1000010 V:
87,9848450049413 km/h
  Current X: 172
  Current Y: 216
  Points: 108
  Start X: 176
  Start Y: 232
  Start frame: 1059
```

```
1140 frames    19 blobs & noise points    Current objects: 1
```

Fig. 19. Tracking subsystem.



Video įėjimo kanalai: [✓ 1 kanalas] 2 kanalas 3 kanalas 4 kanalas

Įėjimo nustatymai: Šviesumas: Kontrastas:

Formato nustatymai: Spalvos: 256 spalvos Rezoliucija: Bet koks kadras

Pažeidimo faktas / informacija apie savininką

Savininkas: Valdas Stakauskas

Asmens kodas: 37409271452

Vairuotojo pažymėjimas: KC046527

Draudimo polisas iki: 2005 04 28

Pažeidimai: 1

Baudos taškai: 4

Automobilis: Ford Sierra Tipas: Lengvasis Spalva: mėlyna

Numerio identifikacija

Rasti numeriai: KLY275

Numeris: KLY275

Numerio aukštis h: 18

Numerio pozicija:

v / kairė:	v / dešinė:	a / kairė:	a / dešinė:
x: 230	x: 340	x: 230	x: 340
y: 83	y: 76	y: 101	y: 94

Papildoma informacija:

K 98	L 98	Y 79	2 90	7 97	5 95
X 17	E 10	V 52	Z 63	2 29	S 38
4 4	C 5	W 3[illegible]	O 41	1 20	E 17
V 1	G 1	1 24	0 37	T 18	6 15
I 1	I 0	I 18	D 27	V 12	3 8

Atpažinimo trukmė: Dabartinis: 47 ms Geriausias: 47 ms Blogiausias: 62 ms Vidutinis: 52 ms

Fig. 20. LPR subsystem.

5 Resume and future works

The study presented explains computer vision based approach to road monitoring and traffic analysis problem. Such tasks as vehicle tracking, speed measurement, jam detection and number plate recognition are considered. Approved methods and algorithms are implemented in the intelligent video monitoring system with data transferring over computer networks and archiving in local and central databases. System implementation confirmed theoretical and design findings, suitable efficiency of proposed methods and algorithms.

Future work will cover complex testing of the system, and more detailed development of modified algorithms. This will include comparison to other (in many cases succesful) methods of computer vision (see [5–7]).

References

1. G. D. Sullivan, K. Baker, et al. Model-based Vehicle Detection and Classification using Orthographic Approximations, in: *Proc. British Machine Vision Association Conference,* 1996.

2. D. A. Forsyth, J. Ponce. *Computer Vision. A Modern Approach,* Prentice Hall, 2003.

3. D. Beymer, et al. A Real-time Computer Vision System for Measuring Traffic Parameters, in: *Proc. IEEE Conf. On Computer Vision and Pattern Recognition,* 1977.

4. L. G. Shapiro, G. C. Stockman. *Computer Vision,* Prentice Hall, 2001.

5. L. A. B. Jähne, H. Haußecker, P. Geißler. *Computer Vision and Applications,* Academic Press, 1999.

6. C. Chui. *Kalman Filtering: with Real-time Applications,* Springer Verlag, 1991.

7. M. West, J. Harrison. *Bayesian Forecasting and Dynamic Models,* Springer Verlag, 1997.