ACQUISITION OF INPUT DATA FOR TRANSPORTATION MODELS

Petr Cenek[1], Peter Tarábek[1]

[1]University of Žilina, Faculty of Management Science and Informatics,
010 26 Žilina, Univerzitná 1, Slovak Republic

Petr.Cenek@fri.uniza.sk (Petr Cenek)

Abstract

Simulations serve to experiment with a model to explore system behaviour and to test and/or design suitable control methods. Simulation model must be correct and supported with accurate input data to achieve these desired goals.

Input data on transportation system represent a substantial part of the simulation model. The input data are unique for each simulated project and that is why an efficient acquisition of input data is an important task. The paper deals with data acquisition methods based on pattern recognition supposed to be suitable for these needs.

The infrastructure data can be acquired from drawn plans and maps or from images. Even if infrastructure might be already available, rarely are complete with all necessary attributes for microsimulation models. Such attributes can be recognised and added to available data sets.

Data on vehicles and transportation flows can be extracted from video sequences of real traffic. Processing of video data seems to be the only effective method to get detailed information on vehicle behaviour in a transportation flow.

Acquisition of input data by pattern recognition methods may deliver complete data for microsimulation models quick and easy.

Keywords: simulation, transportation, pattern recognition, input data acquisition.

Presenting Author's biography

Petr Cenek. Born in Prague, graduated in Mechanical Engineering at the Czech Technical University, Prague. Since 1973 he has worked with University of Žilina (former University of Transport and Communications) in Žilina, Slovakia. His main professional interests are in Computer Science, Optimisation Methods, Simulations and Computer Graphics applied in Transports.

1 Introduction

Transportation systems are usually widespread systems, which may include different transportation modes and may be a part of logistics and production chains. Microscopic models of a transportation system must respect to a certain degree special characteristics of a transportation mode but all the models have similar structure consisting of infrastructure (network), transportation flows and control subsystem. Design and optimisation of the control subsystem should frequently result from simulation experiments and so the main interest in creating a model will be paid to infrastructure and vehicles (transportation flows). Let us shortly discuss used data and desired data characteristics.

The infrastructure (transportation network) consists of nodes and edges of the network. Model of transportation infrastructure defines vehicle trajectory (its kinematics and dynamic characteristics) and that is why some input attributes are vital for further simulation experiments.

At first, infrastructure model must define accurate and unambiguous position of nodes and correct network topology. The unambiguous positioning of nodes is not specially treated in general automatic pattern recognition systems, so it is one of problems to be solved.

Secondly, vehicle kinematics and dynamics depend on road geometry and that is why an accurate edge shape must be estimated. Many models define infrastructure links as a polyline or a spline, which is comfortable for data storage and visualisation but unsuitable for vehicle dynamics estimation. Edge curvatures must be known for calculation of centrifugal acceleration and thus define physical speed limits for vehicles.

Finally, a bare medial axis of an edge is not sufficient for real traffic simulation but a road division to traffic lanes and other traffic signs and regulations must be known. So the last problem is a correct estimation of edge attributes such as shape and lane distribution.

For the mentioned reasons, the infrastructure data for microsimulation models must include three major types of attributes, namely:

- an accurate and unambiguous position of nodes,
- an accurate edge shape (as a smooth curve consisting of straight and curved sections),
- an accurate description of traffic lanes on edges (roads) and in nodes (junctions) of a network.

Some of these attributes may be automatically estimated at the infrastructure recognition process some must be input manually using a dedicated editor.

Vehicle characteristics can be estimated from video sequences of real traffic. Video cameras are already widely used for traffic survey and vehicle counting and so only proper data processing is necessary to yield the desired vehicle characteristics. The data loaded in a microscopic simulation model must correspond to vehicle behaviour in real traffic. It means that these data does not describe only technical characteristics of a vehicle but they must represent behaviour of a vehicle together with a driver as one compound unit.

Dynamic vehicle characteristics depend on its construction and define speed limit, acceleration and deceleration rates of the vehicle. Several types of dynamic vehicle characteristics for microsimulation models can be found in technical specifications for vehicles of a typical fleet proper to a certain country or region. The differences in vehicle fleet composition between various countries become less important with a growth of global markets. Even so, a certain difference exists due to different standards of living in these countries.

A more important difference can be observed in driver's behaviour, which depends strongly on local tradition and driving style. Quite different behaviour can be expected in Norway, in Italy and in a Middle East country. That is why characteristics of vehicle's behaviour cannot be freely imported from one general source but must be estimated for each individual project and for traffic situation in a regional environment. Traffic analysis from realistic video sequences seems to be the only efficient tool to acquire these input data. The traffic analysis is a complex task which includes many problems of projection geometry, pattern recognition and object classification.

2 Recognition of infrastructure

Recognition of transportation infrastructure can be used whenever digital data are not readily available. Data availability means that data are suitable for a simulation model i.e. they are:

- freely accessible (as a commercial product),
- with sufficient accuracy,
- complete with all necessary attributes,
- for acceptable costs.

Otherwise an infrastructure recognition system may help to acquire the data from scratch or to add necessary attributes into incomplete data sets.

The recognition system processes input data according to a generally used scheme composed of following basic steps:

- image segmentation,
- pre-processing (morphological operations),
- thinning and
- vectorisation (post-processing)

to yield desired vector data.

The first three steps are generally known and so the appropriate algorithms only have to be carefully implemented to prepare correct data for vectorisation. The last step (vectorisation) must recognise final infrastructure with precisely defined and positioned nodes and accurate definition of edge shapes. A definition of road lanes is currently performed manually with a major help of infrastructure editor. Let us discuss individual steps of the recognition process.

Segmentation

Analysed image of a transportation infrastructure contains data of interest (infrastructure data) and some background information. Segmentation step separates infrastructure data so that resulting binary image represents infrastructure as black pixels and all other background objects as white pixels.

Infrastructure data can be separated using a decision rule, which can be for example a threshold function defined as follows:

$$g(x,y)=\begin{cases}0, & \text{if} \quad f(x,y)<T\\ 1, & \text{if} \quad f(x,y)\geq T\end{cases}$$

where $g(x,y)$ is the value of a pixel with coordinates x, y in a resulting binary image and $f(x,y)$ is the original value of that pixel. T is a threshold value.

More elaborated rules use several thresholds to separate correctly objects in the image. There are many local and global threshold techniques, but a fully automatic threshold set up method for various drawn maps can be hardly found [6].

Pre-processing

The binary image is processed in pre-processing step to remove imperfections and to amplify desired features of line objects. Accurate pre-processing removes isolated small objects, reduces boundary noise, fills small holes in objects and joins disconnected objects. Binary morphology operators opening and closing provide very good results for this kind of tasks. Operations opening and closing can be combined to remove all mentioned imperfections [7] (of course to a certain degree only).

Fig. 1 illustrates a difference between desired and acquired results after thinning step. Careful pre-processing helps to receive results close to desired ones.

Thinning

The aim of this step is to create skeleton one pixel thick while connectivity, shape and position of the junction points should be preserved. Further, skeleton links should lie in the middle of a shape (medial axis), skeleton should be immune to noise and excessive erosion should be prevented (length of lines and curves preserved).

Thinning algorithms remove outer pixels of a motif layer by layer to produce one pixel thick skeleton. The thinning shows good results for objects of transportation infrastructure (roads), the length of which is much larger than their thickness. The resulting skeleton represents infrastructure as a set of raster points (a binary bitmap file), which must be further processed to yield a vector representation.

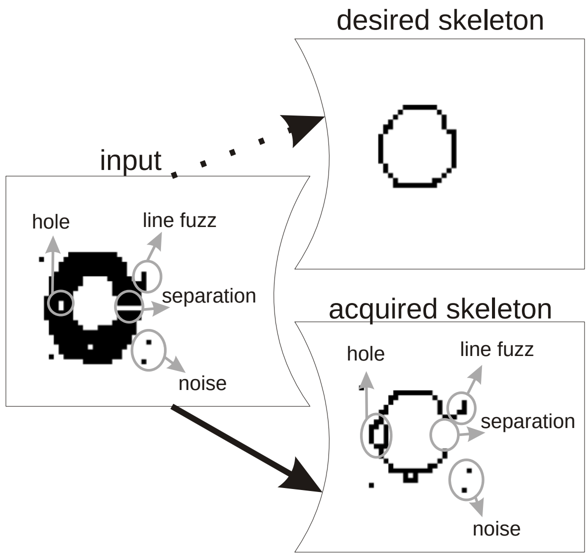

Fig.1 Input image, desired and acquired skeleton

The nature of thinning algorithms can be parallel or sequential. Parallel thinning algorithms make their decisions on deleting pixels using a bitmap from the previous iteration, while sequential algorithms use an actual bitmap.

Raster-to-vector conversion

Vectorization step converts raster data of a skeleton to vectors representing infrastructure. Topology and shape of roads must be preserved. Topology preservation means that junction points and their positions are accurately estimated and skeleton connectivity is retained otherwise false additional junction points would be created. Shape preservation includes mediality, prevention of excessive erosion and immunity to noise.

The topology and shape preservation constitutes the most serious problems, which are rarely solved accurately enough. A new method based on recognition of node candidate clusters has been proposed in [9]. A priority number is calculated for every pixel of the cluster based on the number of 8-neighbors (N8) candidates and the number of 4-neighbors (N4) candidates. Candidate pixel with the largest priority is then selected for a node. Position of nodes in the skeleton is accurately recognised by this method. The difference of local and cluster approach is shown in Fig.2.

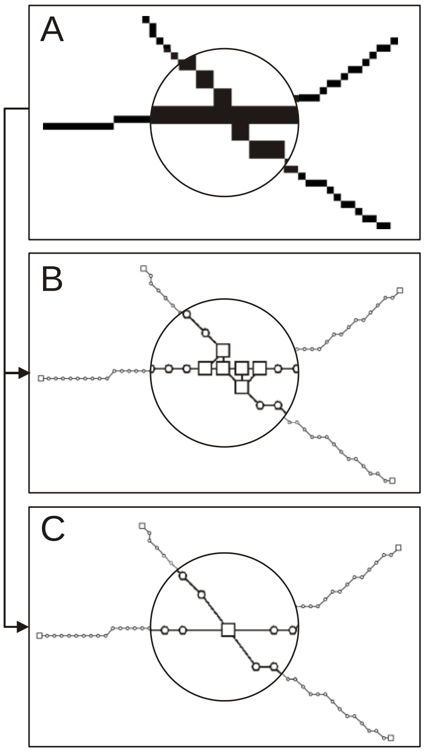

Fig.2 Local approach (B) and cluster approach (C)

Post-processing

The raster-to-vector conversion delivers a set of elementary vectors (as shown in Fig. 2). Even a straight line contains many unnecessary points, which should be eliminated. A polynomial approximation can be used to find straight and curved sections of edges and such approximated links can be then used to estimate accurate position of a node. Recognition of straight lines and arcs is discussed in [8] for example.

Additional post-processing of vector data may include pruning, connecting incorrectly separated objects, improving accuracy of junction points positioning and recognition of further attributes of infrastructure links such as length, width or type (colour).

Polynomial approximation

The approximation of infrastructure links is a decisive step to ensure quality of final results. As mentioned, the recognition of infrastructure links supposes that only straight sections and circular arcs may create a link. So the recognition algorithm is limited to these two types of geometry elements.

Two basic approaches were tried to find individual sections and estimate their attributes:

- inverse interpolation,
- approximation using least squares method.

Although the problem seems to be easy to solve, an accurate estimation of individual sections is hard to attain even for input data of perfect quality (artificially generated). The algorithm works as follows:

Step 0: Set counter I=0

Step 1: From positions of the first 3 points after counter *I* estimate expected attributes of the next geometrical element. Set $I = I+2$

Step 2: Increase the counter and check the point at a position *I*. If it belongs to a recognised element then refine the element attributes using position of the new point and continue repeating *Step 2*. Otherwise continue at *Step 3*.

Step 3: Set index of the final point of the created element to $K=I$ and search for the optimal value of K so that the element fits the input points the best way.

Step 4: Set counter $I=K$ and if there are some remaining points continue at *Step 1* otherwise *End*.

The approximation of roads should be a smooth line, which means that the line and its first-order derivative are continuous as shown in Fig. 3.

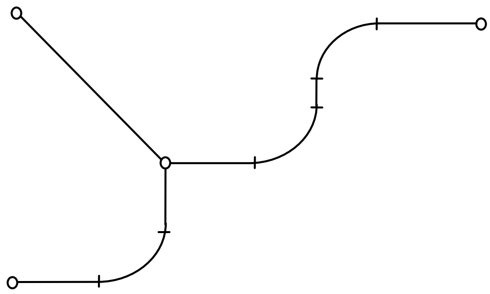

Fig.3 Approximation of road sections

The approximation step recognises individual sections of the line. Resulting infrastructure consists of smooth lines, curvature in curved sections is well defined and accurate position of network nodes can be estimated as an intersection of approximated lines.

3 Estimation of vehicle characteristics

An analysis of video sequences for estimation of vehicle characteristics follows a very similar scheme of data processing as in infrastructure recognition system. A video sequence is divided into individual frames at first. The individual frames are then pre-processed, moving objects are recognised, classified and finally characteristics of individual vehicles can be estimated. Data of individual vehicles at the same traffic situation can be then statistically evaluated and probability distributions of velocity and acceleration can be used to generate vehicles with randomised attributes in a simulation model.

Vehicle is a mobile object in a transportation system. The traffic surveillance or any video registering provides information on mobile objects as units consisting of a vehicle and a driver. Only behaviour of such composite units can be recognized in traffic, which is quite relevant for simulation projects [5].

The acquisition of vehicle characteristics from a video sequence consists of registering video sequence of actual traffic, recognition of vehicles (moving objects), estimation of vehicle positions in the real world co-ordinates (scene geometry), derivation of vehicle characteristics and statistical evaluation.

Vehicle position

Traffic characteristics can be recognised with a fixed camera position and also scene background can be assumed fixed (not changing in time). This allows to find moving objects by subtracting background information from an actual image. Generally, any method can be used to find moving objects usually comparing individual video frames against a background image or against successive frames. Frequently used methods are *background subtraction*, *temporal differencing* (differences between two successive frames) or *optical flow*. These methods can be programmed in proper applications or specialized cameras can offer a pre-processed output with recognition of moving objects.

The information on moving objects is further analysed to estimate vehicle position and other characteristics like speed and/or acceleration. Vehicles are 3-dimensional objects and move in a 3-dimensional space while video data can provide only 2-dimensional pictures so the real world co-ordinates have to be estimated [2]. Vehicles move on a road or (roughly speaking) in a linear co-ordinate system and so speed or acceleration must be estimated in the actual direction of the road.

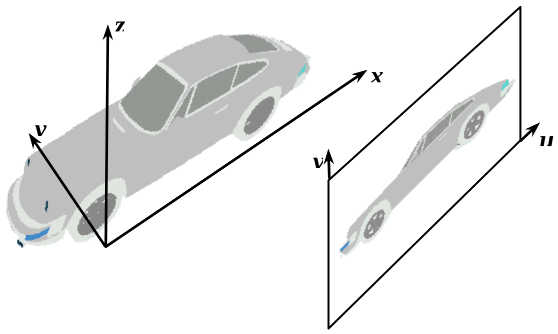

Fig. 4 Vehicle 3D world co-ordinates vs. projection 2D co-ordinates

The problem of a discrepancy between a real word space and video image is illustrated in Fig.4. Let us denote world co-ordinates (*x, y, z*) and co-ordinates of video recording frames (*u*,*v*). The camera delivers a perspective projection defined by a projection matrix ***M***, so that projection co-ordinates (*u*,*v*) can be estimated from world co-ordinates (*x*,*y*,*z*) using transformation formula

$$(u, v, h) \leftarrow (x, y, z, h) * \boldsymbol{M}_{3,4}$$

where (*u, v, h*) and (*x, y, z, h*) denote homogenous co-ordinates (completed by scaling factor *h*). The opposite calculation symbolically denoted as

$$(x, y, z, h) \leftarrow (u, v, h) * \boldsymbol{M}_{4,3}^{-1}$$

cannot be done as ***M*** is not a square matrix and so an inverse matrix does not exist. So some assumptions must be taken to allow world co-ordinates estimation.

A simple assumption is that the scene lies in a plane. World co-ordinates in the scene can be then calculated supposing camera axis is defined by vector ***a*** and normal vector ***n*** defines the scene plane. Main goal of the recognition is to derive functions of velocity and acceleration against time or against vehicle position. The velocity is estimated from differences of vehicle's positions and that is why absolute world coordinates and origin of the co-ordinate system are irrelevant. So origin of the camera co-ordinate system as well as origin of the world co-ordinate system can be placed on the camera axis.

Let us now construct an auxiliary plane parallel to camera image plane (perpendicular to camera axis) at a point ***P*** where camera axis intersects the scene plane as illustrated in Fig.5.

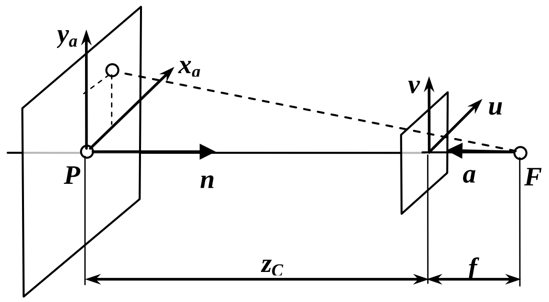

Fig. 5 Co-ordinates on auxiliary plane

Distance from point ***P*** to a camera plane is marked z_C. The centre of perspective projection ***F*** lies also on the camera axis at distance *f*. Co-ordinates x_a, y_a, z_a of points in the auxiliary plane can be estimated using simple relations:

$$x_a = u\,.k$$
$$y_a = v\,.k$$
$$z_a = 0$$
$$k = \frac{z_C + f}{f}$$

The same formulas hold for co-ordinate differences

$$dx_a = du\,.k$$
$$dy_a = dv\,.k$$

and velocity can be calculated from a distance

$$d_a = \sqrt{dx_a^2 + dy_a^2}$$

These relations are valid for a bird's eye view where camera is placed high over a horizontal scene.

Unfortunately, a real scene plane is situated in a general position defined by normal vector ***n*** with components n_x, n_y, n_z as shown in Fig. 6.

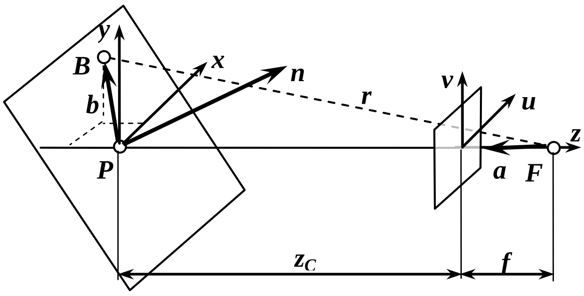

Fig. 6 Projection on a general plane

Relations between co-ordinates must be derived estimating intersections of projection rays with the scene plane. Point ***B*** is an intersection of ray ***r*** with projection plane. Vector ***b*** lies in the projection plane and is defined by a point ***P*** in the plane and point ***B***. Supposing that ***P***≡(*0,0,0*) is the origin of the co-ordination system then ***b*** is a position vector of point ***B***. Vectors ***n*** and ***b*** must be orthogonal and so following equation holds:

$$\boldsymbol{n} \,.\, \boldsymbol{b} = 0 \qquad \text{or}$$

$$\boldsymbol{n} \,.\, \boldsymbol{B} = 0$$

Vector ***r*** is defined by focus ***F***≡(*0,0* z_C+*f*) and a point ***A***≡ (*u,v,*z_C) in the camera plane. Parametric equation for vector ***r*** is then

$$\boldsymbol{r} = \boldsymbol{F} + t.(\boldsymbol{A}\text{-}\boldsymbol{F})$$

Point ***B*** lies on the vector ***r*** so the equation of ray ***r*** may be substituted into condition for orthogonal vectors yielding a formula for estimation of *t*

$$t = \frac{-\boldsymbol{n}.\boldsymbol{F}}{\boldsymbol{n}.(\boldsymbol{A}-\boldsymbol{F})}$$

Co-ordinations of projection point ***B*** can be estimated by substituting value of parameter *t* into equation of vector ***r***. Distances for velocity estimation are then calculated from differences of all three co-ordinates in the world co-ordinate space

$$d = \sqrt{dx^2 + dy^2 + dz^2}$$

Scene calibration

The scene plane normal vector has three components, which together with distances z_C and *f* define the projection. A scene can be calibrated by estimating these parameters. The normal vector can be calculated from positions of 3 landmarks in a scene plane. World co-ordinates of the landmarks can be measured using a GPS system, which gives co-ordinates $\boldsymbol{P_1}=(x_1, y_1, z_1)$, $\boldsymbol{P_2}=(x_2, y_2, z_2)$ and $\boldsymbol{P_3}=(x_3, y_3, z_3)$. The normal vector can be estimated from equation

$$\boldsymbol{n} = \frac{(\boldsymbol{P}_3 - \boldsymbol{P}_1) * (\boldsymbol{P}_2 - \boldsymbol{P}_1)}{\left| (\boldsymbol{P}_3 - \boldsymbol{P}_1) * (\boldsymbol{P}_2 - \boldsymbol{P}_1) \right|}$$

Distance z_C from the scene to the camera can be directly measured or calculated from world co-ordinates of the camera. Distance *f* represents focus length of the camera position. So all projection parameters can be estimated and video recognition system is thus calibrated.

Recognition of 3D vehicles

Unfortunately, vehicles do not lie in the scene plane but they are 3D objects. So the assumption of objects positioned in the scene plane is not fulfilled. The systematic error arising from this fact should be analysed and corrected, if possible. This error depends on a camera position and there is no error for bird's eye view. The lower is camera positioned the bigger error will arise.

Basic reflection can use image of a car from Fig. 4. This image (a blob) can be framed in a polygon, which encompasses the whole vehicle. Further, a prism can be constructed, which encompasses the car in a real 3D space as shown at the right hand side of Fig.7. Projection of the prism centroid ***T*** coincides with median point of the frame. This is an important result as the centroid of a car may be taken as a reference point when evaluating motion of a car.

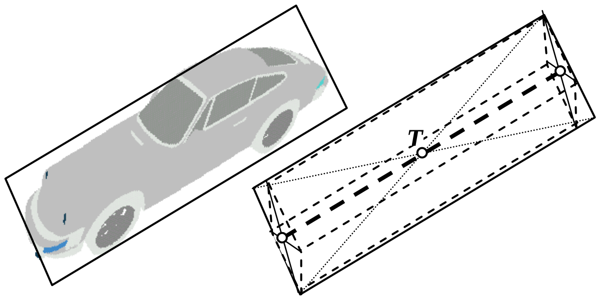

Fig. 7 Projection on a 3D car

Another approach might take the lowest point of the blob as a reference point. In this case finding of such a point is more difficult and the result is more sensitive to any imperfections in the image (position of such a reference point may change due to imperfections in the image much more than position of a median).

4 Conclusions

Presented recognition methods are partially implemented (especially methods for infrastructure recognition), theoretical analysis should serve for estimation of an attainable accuracy of results and/or elimination of systematic errors using suitable corrections.

The recognition of infrastructure is sufficient for practical use, especially the accurate node positioning and recognition of road shapes are valuable results of our research. The only remaining problem, which has not been satisfactory solved yet, is removing of thinning artefacts. In some cases (like thick lines and acute angles) the artefacts are created by thinning algorithms, which would alter topology of the resulting network. So even if the accurate vectorisation algorithm is used these inaccuracies can be transferred into a vector representation.

The performance of the recognition algorithm was improved using original approach which processes just outline pixels. The approximation algorithm delivers straight and curved segments, which is desirable for microsimulation models. Traffic lanes organisation can be edited manually using a specialised editor together with road signs as used in real traffic.

Automatic recognition of visual information about vehicles and traffic flows is based on theoretical reflections, which yielded basic formulas for calculation of world co-ordinates in 3D space, for calibration of the scene and camera. Further research supposes implementation of proposed methods and evaluation of different camera positions and arising errors. Finally, recommendations for scene choice and camera positioning will serve for proper organisation of traffic measurements in real traffic.

5 References

[1] BOSE, B., GRMSON, E.: Ground Plane Rectification by Tracking Moving Objects. Proc. of Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2003

[2] CENEK P., JÁNOŠÍKOVÁ Ľ.: Models and Optimization in Transports and Logistics. EDIS, University of Žilina, Žilina 2010, ISBN 978-80-8070-951-8

[3] GONZALES, R., R. WOODS, R., Eddins, S.: Digital Image Processing. Addison Wesley, 2002

[4] KUMAR, P.: Framework for Real-Time Behavior Interpretation From Traffic Video. IEEE Trans. on Intelligent Transportation Systems, Vol. 6, 1/2005

[5] PURSULA, M.: Simulation of Traffic Systems – An Overview. Journal of Geographic Information and Decision Analysis, vol.3, 1/1999, pp.1-8

[6] LIAO, P. S., CHEN, T. S., CHUNG, P. C.: A fast Algorithm for Multilevel Thresholding, Journal of Information Science and Engineering 17, 2001, pp. 713-727

[7] TARABEK, P.: Morphology image pre-processing for thinning algorithms, Journal of Information, Control and Management Systems, Vol. 5, No.1, 2007, ISSN 1336-1716, pp. 131-138

[8] DOSCH, P., MASINI, G., TOMBRE, K.: Improving Arc Detection in Graphics Recognition, Proc. 15th Int'l Conf. Pattern Recognition (ICPR '00), Vol. 2, 2000, ISBN 0-7695-0750-6, pp. 243-246

[9] TARÁBEK, P.: Dissertation Thesis, Žilinská univerzita, Žilina, 2009

Acknowledgement

The research has been supported by a research grant VEGA 1/0361/10 and application project APVV SK-CZ-0075-09