Acquisition of data for transportation models using automatic analysis of graphical information

Petr Cenek, Peter Tarábek[1],

[1] Department of Transportation Networks, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 1,

01 026 Žilina, Slovakia
Petr.Cenek@fri.uniza.sk
Peter.Tarabek@fri.uniza.sk

Abstract. Transportation models demand a time-consuming data input. Automatic data acquisition using pattern recognition methods is therefore an important and effective step of a modeling process. A transportation model consists of infrastructure (transportation network) and vehicles (traffic flows). Input data for both parts of the model can be excerpted from available graphical data. The infrastructure data can be gained from drawn maps, where linear constructions are featured unanimously and so the maps can be processed to a high degree automatically. Satellite and aerial pictures are less suitable for automatic processing. Data on traffic flows are typically limited to vehicle counting (estimation of traffic flow volume or intensity). Detailed transportation models may recognize motion of individual vehicles and so much more input information is needed. The necessary information for such models can be derived analyzing real traffic video sequences. Both types of data acquisition methods are being developed in our department. Due to the scope of the conference, achieved results and experience in the automatic infrastructure recognition are discussed in the paper.

Drawn maps vectorisation uses a standard recognition process consisting of data pre-processing, segmentation (thresholding), morphological operations, thinning and recognition of network objects (nodes and edges of the network). A new approach was implemented for contour thinning algorithm, which accelerates the processing substantially compared to known thinning algorithms. The recognition of linear objects delivers improved resulting accuracy of identification and positioning of network nodes.

Estimation of dynamic vehicle characteristics from video sequences is less elaborated at the moment. It is based on recognition of moving objects in the scene, estimation of a vehicle position in real world co-ordinates and following calculation of actual vehicle speed and acceleration. Consecutive statistical analysis derives expected values of dynamic parameters, which can be further used as input data for vehicle generators in simulation models.

Keywords: transportation infrastructure, pattern recognition, drawn maps

1 Introduction

Transportation systems are large distributed systems with a rich variety of processes to be controlled. Control and management of transportation systems can be approached on a microscopic level, which works with dynamic models of individual vehicles and transportation flows, or on a macroscopic level when only traffic volumes are assumed and optimization tasks are concerned with finding optimal routes and schedules for transportation processes in the network [1].

Every model of a transportation system must comprise transportation infrastructure, transportation flows, and effective optimization methods or simulation procedures for solving relevant planning and management problems. Such a transportation model has to be loaded with correct input data otherwise optimization and simulation results are of no significance. Even if the model and optimization routines can be used repeatedly, input data must be acquired individually for every traffic environment to customize the model on local conditions. The data acquisition is a costly and time-consuming task and so any automation of the acquisition process can help a lot.

Data on transportation flows and vehicle behaviour can be recognised using video sequences from a camera (see [2], [3]). Transportation flow consists of individual vehicles and their mutual interactions. Individual vehicle characteristics like vehicle position, speed and acceleration describing its dynamic behaviour must be available for detailed transportation models, as accurate data are the basic requisite for obtaining reliable results of simulation experiments.

Data acquisition on vehicle behaviour estimates a possibly accurate vehicle position and derives further characteristics like speed and acceleration along a road section. The speed and acceleration values related to a road section must be estimated for typical traffic situations like free traffic and car following on a straight and curved road sections, on an approach to a road junction etc. The data are then used as input data for vehicle generator in simulation models.
A human operator cannot deliver such accurate data by himself and so technical equipment must be used instead. Speed measurement using radar or laser speedometers delivers an actual speed but can be hardly used to estimate the desired speed against distance function. The automatic analysis of video sequences seems to be the only suitable approach to provide a detailed insight into vehicle behaviour. Vehicle characteristics can be estimated by recognition of vehicles, their position in real world co-ordinates and estimation of their speed and acceleration.
The acquisition of vehicle characteristics from a video sequence is a process, which consists of the following steps:

- video registering of actual traffic,
- recognition of vehicles (moving objects in a scene),
- estimation of a vehicle position and derived vehicle characteristics in the real world co-ordinates (scene geometry),
- statistical evaluation of vehicle characteristics.

The first step is the recognition of a moving object in a scene. Generally, the video frames are compared to successive ones, changes between frames are recognised and evaluated. There are many approaches to find out moving objects. Let's mention several at least:

- background subtraction,
- temporal differencing (differences between two successive frames),
- optical flow.

These methods can be programmed in a dedicated application or specialised cameras offer a pre-processed output with recognition of moving objects.
The recognition of vehicle's behavior has not much importance in GIS systems and so the rest of the paper is devoted to automatic acquisition of infrastructure data. The infrastructure data should describe network topology, which is essential for any transportation model, and information on shape of road sections, which is useful for network visualization and microscopic simulations.

2 Vectorization of drawn maps

Infrastructure input data can be recognized from drawn maps. Line objects which represent transportation network are represented in very similar manner in these maps, which simplifies its recognition. The only information available in raster images is information about position and colour of each pixel. Topology, geometry and attributes defining the infrastructure need to be excerpted from them. Vectorization process using general characteristics of drawn maps and line objects was proposed for the infrastructure recognition. This process consists from 5 stages shown in Fig. 1.

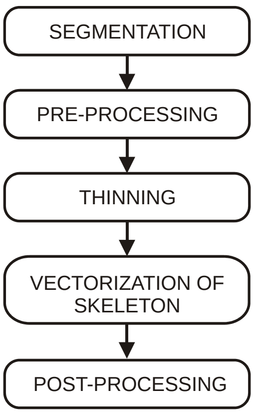

Fig. 1. Proposed vectorization process

High level of automation and generality are main features of the process. An operator is needed only at the beginning of the process when the parameters are setup and at the end of the process for the results verification.

2.1 Segmentation

Before the vectorization process can start, areas of interest need to be separated from others and binary image must be created. Segmentation is used for this purpose. Drawn maps are specific by the fact that they are created artificially for a human user. In drawn maps limited number of colours is used and different types of objects are defined by different, usually easily distinguished colours. These features are appropriate for the threshold method which is used in this step. Threshold values are used to separate line objects from other areas. The result is a binary image where all line objects should be represented as black pixels (foreground) and all other objects including the background as white pixels (background). This reduces amount of data and it also speeds up and simplifies future processing. A general condition for threshold can be 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 pixel with coordinates x, y in resulting binary image, $f(x,y)$ is the original value of pixel and T is a threshold value.
Histogram is used to estimate the threshold values. In ideal case histogram consists of several distinct maximums where each maximum corresponds to exactly one type of object. In real situations multiple threshold values need to be used to correctly separate objects of interest from others. Although thresholding is very simple task and there are many threshold techniques [4], [5], [6] a correct set up of threshold values is difficult.

2.2 Pre-processing

Binary image resulting from segmentation usually contains numbers of small errors and imperfections. They can be caused by compression, antialiasing, bad condition of input map or inaccurate segmentation. In this step a binary image should be improved according to weaknesses of the thinning technique used in the next step. Thinning is sensitive to noise and can produce unacceptable results even if a small noise occur in an image. The most common types of errors are shown in Fig. 2.

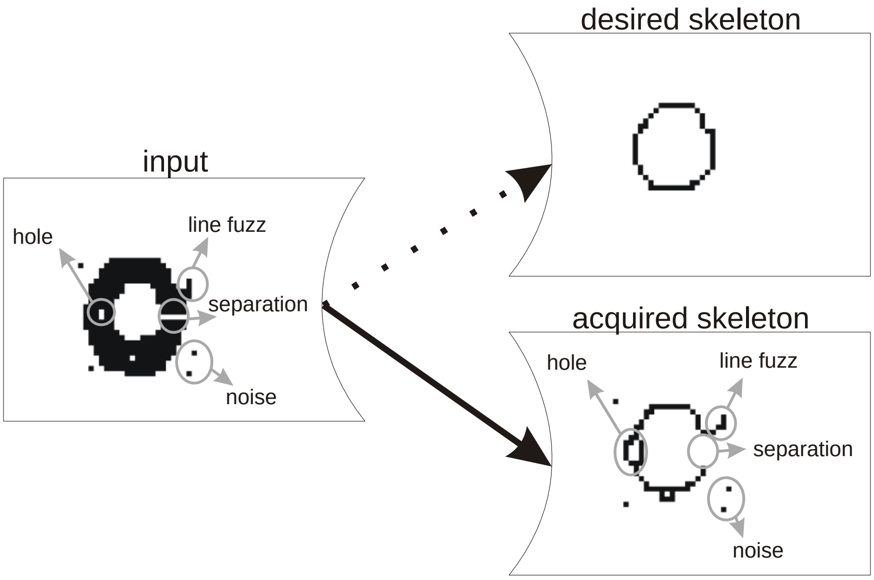

Fig. 2. Input errors and they effect on thinning

In pre-processing a binary image is processed by using the binary morphology operations opening and closing. These operations consist of operations erosion and dilation and they provide very good results for this kind of errors when they are used in combination. For accurate pre-processing number of repetition and correct order of operations need to be set [7].

2.3 Thinning

Pre-processing is followed by thinning. Thinning algorithms remove outer pixels layer by layer in iterative process to produce one pixel thick skeleton. Skeleton contains less foreground pixels while it conserves the main features of the original object. This simplifies future recognition of line objects. The result of thinning algorithms is a modified binary bitmap which must be further processed to yield a vector representation.
Many thinning algorithms were proposed [8]. They usually process all image pixels and majority of them can remove only outer (contour) pixels during the each iteration. Considering the fact that the number of contour pixels in drawn maps usually does not exceed 15% of all pixels in image, large amount of computation time is wasted when all pixels are processed. The contour approach to thinning proposed in [9] is used instead. Its basic idea is to process only contour pixels when conditions for deletion are tested. Although contour approach needs additional resources for recognition, storage and processing of contours, it is still faster then classical approach while it uses similar amount of memory. Contour thinning algorithm was proposed in [10] using Zhang-Suen deletion rules [11]. This algorithm runs approximately 3 times faster than the original algorithm with the same quality of results.

2.4 Skeleton vectorization

In this step, skeleton excepted as raster data is converted into vectors representing infrastructure while topology and shape of line objects must be preserved. Network is then defined by nodes and their position and by edges as vectors connecting nodes. Each skeleton pixel represents either node or vertex which defines the shape of the edge. A local approach where only 3x3 neighborhood of each pixel is inspected when the decision about the type of a pixel is made may produce excessive nodes and edges. These results are inaccurate and need further processing. The cluster vectorization of skeleton proposed in [10] can be recommended for these reasons.
In this algorithm, clusters of candidates for nodes are formed and for each cluster a representative node is found based on the priority number. The priority number is computed as follows:

$$PRIORITY = NNC * 10 + NN_4$$

where NNC is the number of neighbours candidates and NN4 is the number of N4 neighbours candidates (Fig. 3).

N_D	**N_4**	**N_D**
N_4	**P**	**N_4**
N_D	**N_4**	**N_D**

Fig. 3. N_4 and N_D neighbours

The candidate with the highest priority in each cluster is selected to represent the whole cluster. If there are more candidates with the maximum priority number in the same cluster, coordinates of representative node are computed as average of coordinates of all such pixels. Besides the representative node each cluster is defined by its border. All edges touching the border of the cluster are directly connected to the representative node of that cluster. The difference between the local approach and the cluster approach is shown in Fig.4.

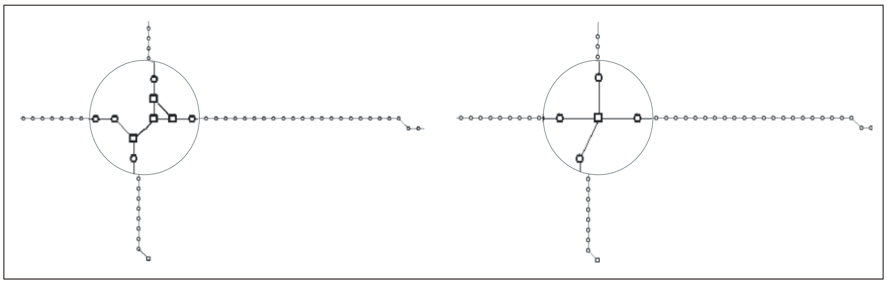

Fig. 4. Local approach (left) and cluster approach (right)

2.5 Post-processing

Post-processing consists of 3 steps:

1. automatic recognition of attributes
2. removal of topological and geometrical inaccuracies
3. polygonal approximation

Automatic recognition of attributes can greatly speed up input data acquisition time. Length and width of edges are 2 main attributes which are recognized in the proposed vectorization process.
Length of the edge is computed as a total sum of Euclidean distances between neighbouring vertexes and between the starting node and the first vertex and between the last vertex and the ending node.
Recognition of width of the edge is started in the thinning step. The thinning algorithm can be modified in order to save information about the iteration in which each pixel was deleted by thinning. This information is used to count the width in place of each skeleton pixel as follows:

$$W(P[i][j]) = round(2.AV[i][j]) + 1$$

where:
- $P[i][j]$ is skeleton pixel with coordinates i, j
- $AV[i][j]$ is average of nonzero iteration numbers estimated from all neighbors of pixel $P[i][j]$ which do not belong to skeleton.

In post-processing the width of the edge is estimated as average W of all skeleton pixels forming the edge.
Additional processing of vector data includes removal of “blind” (or terminal) edges the length of which is less than L_E value. The value L_E is calculated as follows:

$$L_E = \frac{W_A}{2} + p1 * (\max\{\frac{W_A}{2}; n\}) + p2 * n$$

where:
- W_A is average width of incidental edges of the processing edge
- n is a parameter which stands for the size of boundary noise
- $p1$, $p2 \in \{0, 1\}$ are parameters which define the type of the used calculation

Default settings for calculation of L_E are $n = 3$, $p1 = 1$ and $p2 = 0$.
After the vectors are recognized in previous step vector data usually contain too many vertices which can be reduced by some kind of polygonal approximation [12], [13]. Also straight lines and arcs can be recognized in this step [14].
Results of proposed vectorization process are shown in Fig.5 and Fig.6.

Fig. 5. Result of proposed vectorization process

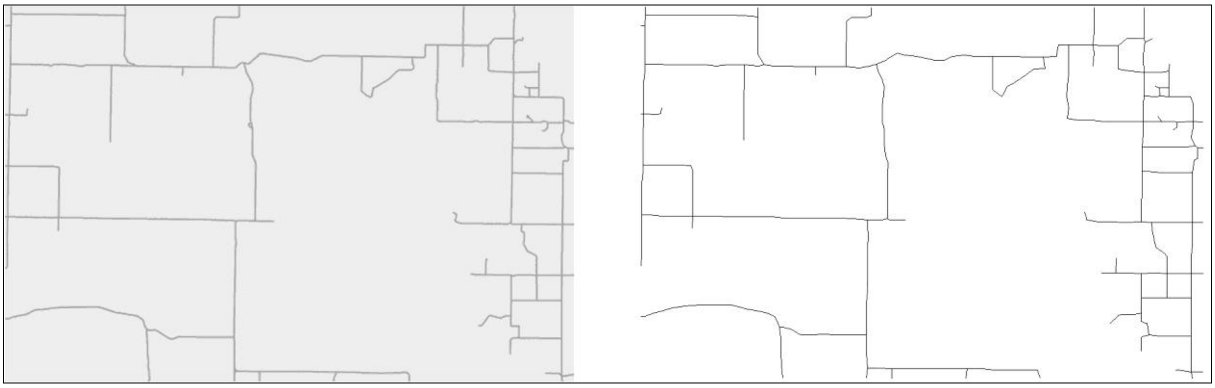

Fig. 6. Result of proposed vectorization process

3 Conclusions

A new method of transportation network recognition from drawn maps is discussed in the paper. A precise definition of nodes and edges of a network is considered to be the most important criterion on quality of the recognition process because they are vital for any further usage of the acquired data in optimization and simulation experiments.

The proposed method corresponds to a general scheme of a pattern recognition process, nevertheless, the attained acceleration of the thinning process and especially the improved quality of the recognized network data can be seen as a significant improvement. The improved quality of the recognition process can be seen in a precise definition of road junctions and their position in recognized network, which is significantly better compared to results obtained by local approach often used in recognition routines [10] as shown in Fig. 4. Skeleton vectorization is completed in one step and the resulting network accurately matches the skeleton obtained by thinning. The morphological operations allow eliminate disturbing inscriptions in the underlying graphics as shown in Fig.5. The method was implemented and tested under BORLAND DELHI environment. The acquired results as shown in Fig.5 and Fig.6 fully comply with the requirements of transportation systems modeling.

Further research may focus on eliminating the thinning artifacts which are often produced by thinning and can alter the topology of recognized network. Also inscriptions like street name, mileage data etc instead of being just eliminated, can be recognized as alphanumeric data and used up as another attribute in the recognized infrastructure (an edge attribute).

Reference

1. CENEK, P.: Models and Optimization in Transports and Logistics. EDIS, Žilinská univerzita, 2008
2. HLAVÁČ, V., ŠONKA, M.: Computer vision. In Czech Počítačové vidění. Grada, Prague, 1992, ISBN 80 85 424 67 3
3. HOOSE, N.: Computer Image Processing in Traffic Engineering. John Wiley New York, 1991, ISBN 0 471 92974 3
4. BOBIN, J., STARCK, J.-L., FADILI, J., MOUDDEN, Y., DONOHO, D.L.: Morphological Component Analysis: an adaptative thresholding strategy, IEEE Transactions on Image Processing, Vol. 16, No. 11, 2007, ISSN 1057-7149.
5. OTSU, N.: A threshold selection method from gray-level histogram, IEEE Transactions on System, Man and Cybernetics, Vol. 9, No. 1, 1979, ISSN 0018-9472.
6. LIAO, P. S., CHEN, T. S., CHUNG, P. C.: A fast Algorithm for Multilevel Thresholding, Journal of Information Science and Engineering 17, 2001.
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.
8. LAM, L., LEE, S., SUEN, C. Y.: Thinning Methodologies-A Comprehensive Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, 1992, ISSN 0162-8828.
9. TARABEK, P.: A Contour Approach to Thinning Algorithms, Communications - Scientific Letters of the University of Zilina, Vol. 11, 3/2009, ISSN 1335-4205.
10. TARABEK, P.: Dissertation Thesis, Žilinská univerzita, Žilina, 2009.
11. ZHANG, T. Y., SUEN, C. Y.: A fast parallel algorithm for thinning digital patterns, Commun. ACM, 27(3), 1984, ISSN 0001-0782.
12. KOLESNIKOV, A., FRANTI, P.: Data Reduction of Large Vector Graphics, Research Report, A-2003-2, Department of Computer Science, University of Joensuu, Finland, 2003, ISBN 952-458-311-9.
13. SARFRAZ, M.: Optimal Curve Fitting to Digital Data, International Journal of WSCG, Vol. 11, No. 1, 2003, ISSN 1213-6972.
14. 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.

Acknowledgment: This work was supported by project “Centre of excellence for systems and services of intelligent transport” ITMS 26220120028

Operačný program VÝSKUM a VÝVOJ

ERDF – European Regional Development Fund

Project Part-Financed by the European Union