Automatic raster-vector conversion of topographic maps

Ph.D. Dissertation theses

Rudolf Szendrei

Supervisor: Dr. István Elek

Eötvös Loránd University
Faculty of Informatics

Doctoral School of Informatics
Prof. András Benczúr, D.Sc.

Doctoral program of Information Systems
Prof. András Benczúr, D.Sc.

Budapest, 2014.

1 Introduction

This thesis introduces a newly developed map interpretation method to automatize the raster-vector conversion process of paper-based large scale Hungarian topographic maps. The basis of this thesis was the IRIS project [4], in the second phase of the development of which I participated. The results of the [7] PhD thesis also provided me with some useful information in this topic. The main goal was to replace the manual vectorization of topographic maps with informatic solutions, or at least to support the vectorization procedure up to an expert level. The concept is that human interactions are reduced and limited to the activities of fine-tuning the software settings for each desired map groups. The main idea is that using the knowledge and map-reading the ways of a human expert can help to automatically separate logical layers of colourful topographic maps (see Fig. 1, where in a top-down way can be seen the point-like symbol layer, the network layer, the surface layer, and finally the map itself made up from these layers). The same idea was also used to deal with the anomaly which arises when an object of an upper layer masks an object of an underlying layer. The program generates as a result one vectorgraphic file for each point- and line-like symbol object type and for the surface object types.

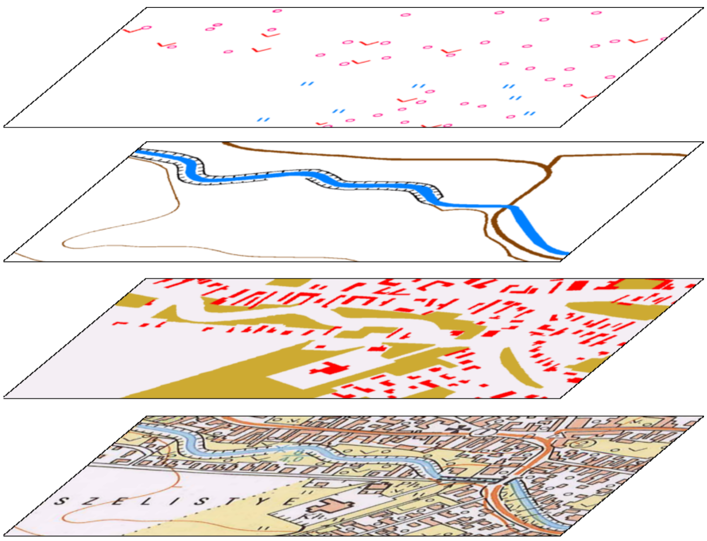

Figure 1: The logic layer hierarchy of topographic maps

This thesis consists of special processing methods given to each map layer (point-like symbol, network and surface layers), a knowledge-based rule and constraint description method and a new algorithm, which is able to automatically parallelize most digital image filter algorithms.

The introduced map interpretation method is needed because the old Hungarian topographic maps have not been vectorized yet. Moreover, most developing countries have paper-based maps in offices waiting for vectorization. Since the created software has a modular structure, later it can be extended with new plug-ins (e.g. import and export modules, image filters, additional languages etc.). This property and the knowledge representation offer the opportunity that the topographic maps of other countries can also be vectorized with the same software.

2 Knowledge-based map interpretation approach

The currently used software and hardware make the manual vectorization task easier, but they do not deal with the map interpretation. Some of these systems help only with the manual vectorization, while others try to produce a vector data model by pure image processing. Since the depiction of topographic maps may vary from country to country, and moreover some spatial relations can be deduced only by human experts, these methods are insufficient to interpret a Hungarian topographic map.

In my first thesis, the former, simple digital image filters-based concept is replaced with a knowledge-based approach [1, 2, 3] to achieve higher efficiency [16]. In this approach, map layers and objects are recognized in the opposite order of the way they were printed on the paper.

Map symbols (point-like symbols, line-like symbols and surfaces) were grouped into three categories by their recognition algorithms. Rule sets were defined for the interpretation of each layer and evaluation algorithms were implemented for each rule type. The main idea of the method is that map making defines a well-defined order of how the map layers are printed onto the paper. The objects of a later printed layer may cover and mask the objects of the formerly printed layers. After the removal of upper layers, visually missing parts can be seen on the lower layers. A set of rules based on the complex map interpretation knowledge try to fill these gaps.

3 Recognition of point-like symbols

Point-like symbols are small objects, which can be found in the book of symbols. In general, point-like symbols also appear in the legend [10]. Their key property is that they appear on the map undistorted, though they may be rotated. My second thesis is an efficient point-like symbol recognition method [11, 12, 15], which is capable of using the special image properties of maps to do a rotation invariant pattern matching, while it also linearizes the recognition time. As an advantage, my method is also able to recognize surface textures, since texture is nothing more than a point-like object repeated a number of times.

The idea of the algorithm is to do edge detection on the map and omit those map points where pattern matching is unnecessary. The edge detection is also needed to determine which pixel of a symbol should be matched to the given edge pixel of the map. An Otsu-thresholding is made on the result of the edge detector, which dramatically reduces the amount of potential edge pixels. This causes that the number of potential pattern matching steps can be largely limited.

The method was compared to a rotation-invariant pattern matching method [17] based upon colour ring-projection, which produced a high recognition rate (about 95%). This method gives a runtime of 261 seconds in the case of a multi-threaded implementation, while the new parallelized algorithm takes only 2-3 seconds on the same machine with a recognition rate > 99%. In practice, the point-like symbols of a highly detailed topographic map can be vectorized manually in approx. 1 hour. The linear runtime was proven in the [11] paper.

4 Recognition of line-like symbols

In the case of line-like symbols, attributes are represented as colours and small line-like drawings printed onto the path of the line-like object or onto the boundary of a polygon object. The recognition of these elements is one the most difficult tasks of the raster-vector conversion, because line-like symbols are often complex drawings. On a map, different line-like symbols may differ only in scale, while they can connect to each other or cross one another. Parallel lines on maps causes also difficulty, because it can be difficult even for a human expert to differentiate the two situations, where in the first case parallel lines belong to one object, while in the second case the parallel lines belong to different objects.

The anomalies described above were beyond of the goals of my thesis and I dealt only with those kinds of line-like symbols where it is assumed that they do not cross each other, and their drawings are continuous.

A classic way of line-like symbol vectorization is introduced in [6], where cadastral maps in binary raster image format are vectorized. Another paper [18] introduces an automatized way of isoline interpretation of topographic maps. Because of the additional features of colour topographic maps, such as road width, capacity, coating etc. cannot be recognized in the classical way [8], a new method [14] has been developed in my third thesis. First, the procedure creates the skeleton of the filtered and binarized raster image of the map and the line-like symbols. Then, it searches for the special points of the skeleton (fork and end points). These points are well known in the fingerprint recognition systems, where they are called minutiae points. In the fingerprint, a fork point in the pattern is an end point in the complementer pattern, so in practice only one type of them is used for fingerprint based identification.

It is important to note that the mentioned complement property is not given for maps, so both the fork and end points are required.

Taking the skeleton of an object as a graph, and labelling corresponding vertices as F (fork) or E (end), an EF graph can be defined. Doing the labelling on the graphs created from the skeletons of the map and the line-like symbols, the structures of the smallest units of each line-like symbol (also called kernel) can be searched inside the EF graph of the map. Furthermore, additional properties can be assigned to the EF graph of each line-like symbol unit. For example, the distance of the adjacent special points can be used as weights in the graph. Using path attribute labels on the path of the symbol, a cyclic property can be also defined for a line-like symbol, if the first and last path points are marked.

Refining the results, the method uses colour indices of pixels of the filtered image for a more accurate processing. The advantage of the method is that it has a linear runtime when it is implemented with deterministic finite state machines. At the end of the process, exported vector data are generalized [5].

The new EF graph concept made it possible to recognize the colour, structure and additional properties of line-like symbols and to retrieve objects from the vectorized data based upon the above properties.

5 Recognition of homogeneous surfaces

During the recognition of homogenous map surfaces, a common task is to join the visually separated surfaces of the same type and to eliminate the map errors. My fourth thesis introduces a solution, where the recognition of surfaces is performed using a mask that describes which points are belonging to the surface layer of the map. The mask can be given e.g. by the remaining map surface after the removal of areas lying under the recognized point-like and line-like symbols. Another method can be when, after the colour segmentation of the map image, the collection of surface colour indexed points are used as a mask. The corresponding colour index of points belonging to the mask is set to the proper surface index, while the colour index of other points becomes undefined. To get a correct result, the two methods (masking and error correction) are used together. In order to recognize surfaces and polygonize them in good quality, a heuristic rule collection was created. With the help of the rules it is possible to describe how the same and different surface types "affect" the prospective type of the undefined surface that lays near to them. In other words, a dominance order can be defined among the known surface types.

The method gives the possibility to join the undefined typed surface not only to one dominant surface neighbour, but it can split the region automatically based on the dominance of the neighbours.

I implemented the proper algorithms to evaluate the rules and the related filters with an easy to use graphical user interface. Furthermore, I made an XML based format, which supports the rule description. This XML format is used to save the given rules in order to use them later in the batched conversion.

6 Automatic parallelization of raster image filters

Nowadays the signal processing speed of digital imaging sensors is increasing, while the size of acquired images is growing exponentially. Although the computational capacity sufficient to process the acquired images, former algorithms remain slow, because those can not efficiently utilize the computing capacities. One reason for the inefficiency is that these algorithms were implemented sequentially. While several articles and books were written about image processing algorithms parallelization, those works are complicated, and they introduce only special parallelized image filters.

My fifth thesis introduces a new method [13], which is able to automatically parallelize the algorithm of most sequential raster image filters. This method is based upon the logical image partitioning, which can be used if an image filter processes the image pixels in a well-defined order, e.g. row-wise, block-wise or block-based recursively. The difficulties of parallel programming of digital image filters come from the image processing method, where neighbouring pixels are also used to calculate the value of the result pixel. In such case, the disjunct partitioning of the result image cannot solve the problem that processes can read the same pixels at the region boundaries at the same time, so overlapping regions should not be read and evaluated in parallel. These regions produce an overhead during parallel image processing, because they can only be processed sequentially, however this cost is negligible in good circumstances. My method deals automatically with the necessary critical sections and minimizes the overhead too.

Because parallel programming requires more intellectual effort and time capacity, this solution efficiently reduces both of these. Furthermore, it improves the quality of the code by eliminating possible parallel programming mistakes; debugging them is often really difficult or almost impossible. Using this method, the speed of the parallelized image filters is in asymptotically linear relation to the number of cores of the computer processor. In this estimation it is assumed that the cost of critical sections are negligible, because the size of the image part processed in the critical section tends to zero compared to the size of the image.

7 Conclusion

A map interpretation method was developed as a combination of five new methods. This interpretation concept simulates better the knowledge and logic of cartographic experts than the former approaches. The applied logic is described by a customizable set of rules and the evaluation algorithms of the rules were also implemented. The description of the rules was in the widely used XML language. XML made it also easy to change the settings and rules both in and outside of the program. An additional advantage of the XML language is that the developed map interpretation system can be expanded with further digital image filters in the future. Thanks to XML, the unknown parameters of the new image filters can be skipped safely, while it remains possible to store or pass them.

It was succeeded to implement the recognition algorithms of point-like symbols, line-like symbols and surfaces with linear runtimes. Additionally, to implement faster the image filters and parallelize them safely and efficiently, an automatic raster image filter parallelization technique was developed for the general sequential image filter algorithms.

The linear runtime algorithms and automatic image filter parallelization-based topographic map interpretation system is faster and more efficient than the former systems. The additional advantage of the automatic image filter parallelization method is that it can be used also in other image processing systems to achieve better performance.

References

[1] Baik, S., et al.: A naive geography analyst system with cognitive support of imagery exploitation. Monroy, R., et al., editor, Lecture Notes in Artificial Intelligence, No. 2974, pp. 40–48, 2004.

[2] Chen, H., et al.: A geographic knowledge representation system for multimedia geospatial retrieval and analysis. International Journal on Digital Libraries, No. 1, Vol. 2, pp. 132–152, 1997.

[3] Corner, R.J.: Knowledge Representation in Geographic Information Systems. Ph.D. thesis, Curtin University of Technology, 1999.

[4] Dezső, B., Elek, I., Máriás, Zs.: Image processing methods in raster-vector conversion of topographic maps. Karras, A. D., et al., editor, Proceedings of the 2009 International Conference on Artificial Intelligence and Pattern Recognition, pp. 83–86, 2009.

[5] Douglas, D. and Peucker, T.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. The Canadian Cartographer, 10(2), pp. 112–122, 1973.

[6] Janssen, R. D. T. és Vossepoel, A. M.: Adaptive vectorization of line drawing images. Computer Vision and Image Understanding, 65(1), pp. 38–56, 1997.

[7] Katona, E.: Automatikus térkép-interpretáció, Szegedi Tudományegyetem, 2000.

[8] Liang, S., Chen, W.: Extraction of line feature in binary images. IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences E91A, 1890–1897, issue 8., 2008

[9] Mayer, H., Heipke, C., Maderlechner, G.: Knowledge-based interpretation of scanned large-scale maps using multi-level modeling. International Archives of Photogrammetry and Remote Sensing, 29 B3, pp. 578–585. 1992.

[10] Samet H., Soffer A.: MAGELLAN: Map Acquisiton of GEographic Labels by Legend ANalysis. International Journal on Document Analysis and Recognition, Vol. 1, pp. 89–101. 1998.

[11] Szendrei, R., Fekete, I., and Elek, I.: Texture based recognition of topographic map symbols. In Karras, A. D., et al., editor, Proceedings of the 2009 International Conference on Artificial Intelligence and Pattern Recognition, AIPR-09, pp. 7–10, ISBN: 978-1-60651-007-0, ISRST. 2009.

[12] Szendrei, R., Elek, I.: Automatic symbol recognition for topographic maps In Geomatics - Riga, pp. 42–48, 2009.

[13] Szendrei, R.: Automatic parallelization of raster image filters. In Majkic, Z., et al., editor, Proceedings of the 2010 International Conference on Artificial Intelligence and Pattern Recognition, AIPR-10, pp. 30–33, ISBN: 978-1-60651-015-5, ISRST. 2010.

[14] Szendrei, R., Elek, I., Márton, M.: Graph-based feature recognition of line-like topographic map symbols. Lecture Notes in Computer Science 6729, pp. 291–298, 2011.

[15] Szendrei, R., Elek, I., Fekete, I.: Automatic recognition of topographic map symbols based on their textures. Lecture Notes in Computer Science 6729, pp. 299–306, 2011.

[16] Szendrei, R., Fekete, I., Elek, I., Márton, M.: A knowledge-based approach to raster-vector conversion of large scale topographic maps. Acta Cybernetica 20, pp. 145–165, 2011.

[17] Tsai, D., Tsai, Y.: Rotation-invariant pattern matching with color ring-projection. Pattern Recognition, 35(1), pp. 131–141, 2002.

[18] Xin, D., Zhou, X., Zheng, H.: Contour Line Extraction from Paper-based Topographic Maps, Journal of Information and Computing Science Vol. 1, No. 5, pp. 275–283, 2006.