

Automatic/Interactive Interpretation of Color Map Images

S.Ablameyko, V.Bereishik, M.Homenko, N.Paramonova, O.Patsko

Institute of Engineering Cybernetics of the National Academy of Sciences

Surganov str., 6, 220012 Minsk, Belarus

E-mail: abl@newman.bas-net.by

Fax: +375 (17) 231-84-03

Abstract

In this paper, a combined automatic/interactive technology and algorithms for interpretation of color maps is proposed. Main stages of technology are: color image binarization, binary image vectorization and vectorized image object interpretation. Special scheme for automatic image vectorization as well as automatic object recognition techniques performed under operator control are proposed. Practical experience in realising this technology is shown.

1 Introduction

Map digitizing by scanning is an important task to input cartographical information in Geographical Information Systems (GIS). This task can be roughly divided into two main stages: raster-to-vector transformation of a raster image with aim to obtain its structural representation, and automatic/interactive recognition of cartographic objects to obtain a required final map representation [1-3].

There are known many systems to perform raster-to-vector (r-t-v) transformation including commercial ones although automatic recognition of cartographic objects is still research task and there are only research papers devoted to this task [2,4,5]. We also have a big experience trying to recognize cartographic objects. We were quite successful in recognition of line objects (isolines, roads, etc.), area objects (forests, cities, etc.) [9]. However, our long time experience shows that it is impossible to recognize automatically a whole map because it was produced to read by human and include objects with different font, orientation, size, etc. Moreover, automatic object recognition is usually followed by a significant amount of interactive post-processing to correct recognition errors. That is why the developed and used at market systems are mainly intended to vectorize map-drawing and represent them in terms of simple enough primitives [9].

In this paper, we describe a developed system and technology for color map interpretation that includes together with automatic vectorization of map layers and

automatic object recognition also automated object interpretation under operator control. Such combination allows one to minimize quantity of digitization errors, decrease time of map digitizing and complexity of operator work. It allows one to reach an acceptable ratio between quality, time and automatization level of a map digitizing process. It also allows one to make object interpretation under operator control instead of automatic image recognition and then correcting recognition errors. Another advantage of the developed technology is that it allows to digitize large-size maps under limited computer memory resources. Inside of the technology, we developed a set of algorithms that could be divided into three groups: image binarization, vectorization, and interactive interpretation.

2 System overview

As input, two types of information are considered: colour maps and map overlay. To represent a cartographical object in output data base, it is necessary to determine its three main parameters: object code, metrics (geometrical coordinates), and semantics (object characteristics) [9].

The main goal of the system is to automatize the process of creation of digital maps on the base of scanned color and black-and-white images. The system includes three main modules:

- **Binarization module** is raster oriented module designed to process color map drawings and to separate color image into several binary images corresponding to different color layers;

- **Vectorization module** is binary image processing oriented module designed to improve quality, transform and perform raster-to-vector conversion of one or more binary images;

- **Digitization module** is vector oriented module designed to digitize map objects and their relations on the base of vectorized data with/without using of raster data.

The common technological basis of these modules is that they are built on Microsoft Windows system and they use coordinated data for information exchange. For the information exchange between modules so-called raster-to-vector projects (**r2v-projects**) are

implemented. They contain raster and vector data represented in internal format. Such projects can be used as output of binarization module and input of vectorization or digitization ones, as output of vectorization module and input of digitization module. Such projects can also be used as intermediate archive for partially binarized, vectorized or digitized data, i.e. as input-output of corresponding modules. In accordance with the state of processing there are *partially* or *entirely binarized*, *vectorized* and *digitized* r2v-projects. The brief scheme of the system and data exchange is shown in Fig. 1.

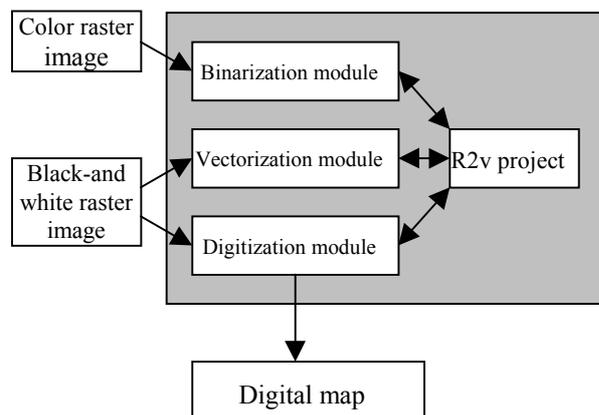


Fig.1. Structure of the system and scheme of data exchange between its modules.

3 Color image binarization

Binarization module takes a scanned colour raster image as input in formats BMP, PCX, and TIFF. Input image can be represented by 8- and 24- bit per pixel. Output of binarization module is layers set together with prepared color image in the form of binarized r2v-project.

At the stage of preprocessing, statistic data (color histograms, pixel neighbourhood relationships, etc.) in RGB space are calculated. Then, initial image is converted into prepared color image in HLS space that is used for viewing and further processing.

Map color layers are extracted automatically. At the same time user has possibility to manage quantity of automatically extracted colors. Such color extraction is based on color space clustering. An interactive correction of color layers is provided by adding/deleting of colors to/from selected layer on the base of clustering analysis. Example of automatic color image processing with extracted color layers is shown in Fig.2 (initial image), Fig.3 (processed image). Extracted color layers are recorded as set of binary images.

Maximal quantity of color layers that can be extracted (using automatic or interactive tools) is not exceed 15.

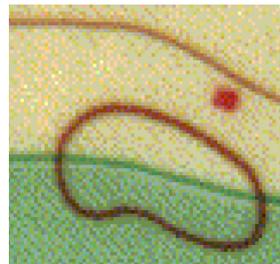


Fig.2. Initial image.

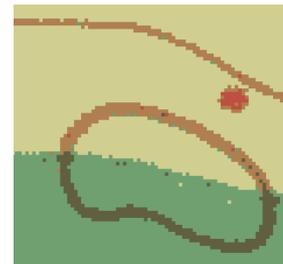


Fig.3. Automatically processed image

In order to process color images with low quality there is possibility to use interactive tools for color layer extraction. Example of interactive color image processing with extracted layers is shown in Fig.4 (initial image), Fig.5 (processed image).



Fig.4. Initial image.

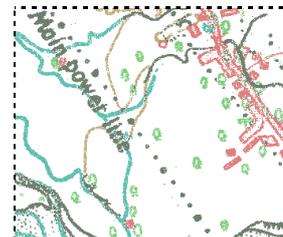


Fig.5. Interactively extracted color layers.

4 Image vectorization

The main operations of this stage are image thinning or contouring depending on the required output object representation, transformation in a vector form with segments and their characteristics extraction and forming an intermediate vector data base for further object recognition.

To process different map-drawing image types, the following variants of raster-to-vector transform have been developed:

- a) Line-like layer vectorization variant, which produces a skeleton description of line objects and symbols.
- b) Region-like layer vectorization variant, which produces a contour description of objects.

To thin image objects, $w+3$ lines are stored in the line stripe, where w is a maximal thickness of objects. Thinning is performed from the first line (bottom) to the last (top) line. Every line in the stripe is thinned different amount of times. The lowest line is completely unthinned and the highest one is fully thinned, all middle lines are partially thinned. For the thinning of one line two adjacent lines (lower and upper) are used. Under pixel analysis a mask is formed which represents a 8-bit code. Using this mask an input in a table is performed, where a new pixel value is defined. A labeling of each full thinned pixel is

performed, and the label is equal to the number of iterations performed.

Thinned image vectorization is performed in one scan and storing three lines in the stripe. In the stripe medial line, black pixels are analyzed using a Crossing number. Depending on the Crossing number value, the following situations are extracted: object beginning, end of object, continuation, merging, splitting, node, and an isolated point. In the vectorization result, the segments of objects bounded by the end points and nodes are extracted. The simple approximation is simultaneously performed.

Contouring algorithm stores two lines in the stripe and solves the following main tasks: extracting the situation in the stripe and its resolution. The possible situations on the image are: object beginning, continuation, splitting, merging, end of object. Under the contour extraction, special buffers are reserved. They are intended for the assembly and storing information about every contour. Then the buffers can be merged or split depending on the processed situation. Simultaneously with the object contour extraction, geometrical parameters of objects are calculated. The relations between processed objects (e.g. object is located inside another one) are computed. Then, vectorization of thin-line images into vector form is performed. In the result of vectorization, an intermediate vector data base is formed which is based on three main element types: connected components (CC), segments bounded by feature points and feature points (nodes and end points) [6]. Example of vectorized map is given in Fig.6.

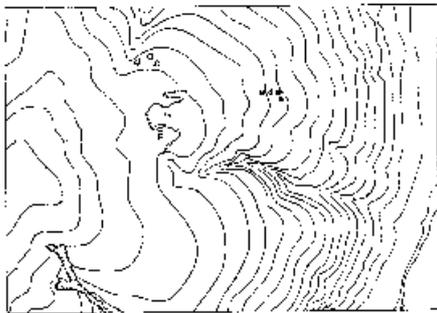


Fig.6. Example of a vectorized map layer.

5 Automatic dashed line detection

The aim of this stage is to assembly line objects having simple structure and consisting from many segments. Typical representatives of such objects are dashed lines. There are known several algorithms to detect dashed lines [7] and there was held a dashed line detection contest [8]. Our algorithm [9] differs from the existed ones by deeper analysis of situations to merge segments that leads to more reliable result in the output. It was developed specially to process complex situations in maps where dashed lines can have various relationships and segments

could be situated very close to each other. The developed algorithm is based on using a dashed line etalons where the etalon for each dashed line includes the following parameters:

- minimal and maximal width and length of segments,
- minimal and maximal distance between neighboring segments (gap),
- minimal available quantity of segments in a dashed line.

6 Interactive image interpretation

Realizing this stage we tried to eliminate drawbacks of traditional manual digitizing process. One of its main drawbacks is a difficult visual control of the digitizing process. User cannot practically control which objects have been already digitized and ones are not digitized yet. In our technology this problem is solved in such a way: not digitized data are displayed on a screen by dim colors and data already digitized by bright colors. User can use many tools for comfortable work (raster background, "microscope", moving "glass", changing pick square, etc.). These tools make the digitizing process more comfortable and less routine.

Another drawback of traditional manual process is difficulty of digitizing elongated twist objects that is usually realized by using follow glass technique. In the system based on automatically vectorized data there are some possibilities for this drawback liquidation. In the result of raster-to-vector conversion the coordinates of such objects parts (segments bounded by special points) are formed automatically. To digitize elongated objects consisted from a set of segments, user can only indicate segments in some order by using more simple technique ('mouse'). Such work simplification makes a map digitizing more faster and less difficult. Furthermore, we incorporated some techniques of automatic recognition (seeking the best chain continuation described above) in this technology. If an operator is sure in correct chain continuation, he must only confirm decisions of the system - it is much more simpler than the digitizing using glass. Several modes of elongated object digitizing have been realized in our technology. For the simplicity we call them as "Draw", "Pick", "Go", "Run", and "Jump" modes.

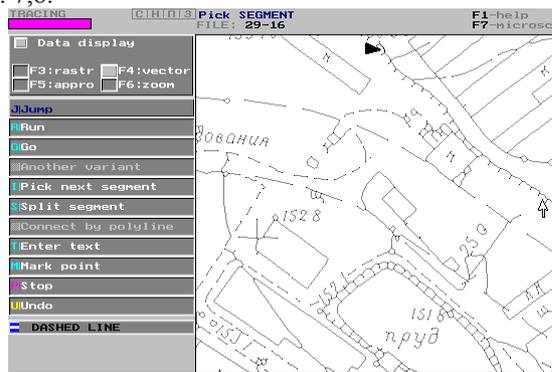
The "Draw" mode permits to input object coordinates by using mouse and is mainly used to digitize text, symbols, bad scanned lines and regions. User can use raster background that is very useful in this mode.

The "Pick" mode permits to pick segments that should be continued by using 'mouse'. The system automatically defines the manner of connection chain tail and picked segment and proposes this variant of chain continuation to user. Choice of connection manner is performed by using already existed connections of segment and distance between two points.

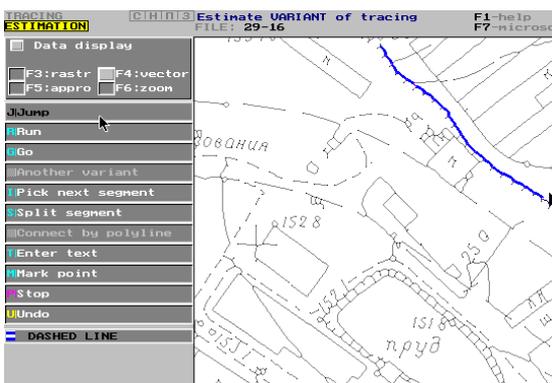
In the "Go" mode, the system extracts some variants to continue object tracing, ranks them by using the above mentioned connectivity criteria and proposes the best variant to user. Data for criteria are calculated by using the given etalon; confirmation/refutation of variants is not made: the system chooses five (or less) variants by using the calculated values. If the chosen variant is not the best one from an user point of view, he can look through other variants or indicate the required segment by using the 'Pick' mode.

In the "Run" mode, the system tries to join segments in a chain during maximum 15 steps automatically if it is sure in their correct merging (connectivity criteria has a biggest value for the best variant and small values for another variants). Otherwise the system passes to the "Go" mode and asks operator's confirmation.

The "Jump" mode is a combination of the "Pick" and "Run" modes: the system finds an optimal way of object tracing from segments and gaps, connects two picked segments and proposes it to user. This mode is very effective for objects that have many connected (by nodes) segments and not very many gaps. Example of tracing line object in a vectorized map in "Jump" mode is shown in Fig. 7,8.



a)



b)

Fig.7. Result of "Jump" operation implementation
a) current chain tail is shown by triangle, picked segment shown by 'mouse', b) optimal way is shown by thick line.

7 Conclusion

The main distinctions of our approach from those already existed are:

- combination of different technological processes to digitize maps what allows to obtain acceptable time characteristics and quite high level of automatization;
- effective algorithms for color binarization;
- specialized image vectorization scheme and original raster- to-vector transform techniques, which allow to digitize large-size maps in a restricted computer memory;
- effective interactive object digitizing techniques.

It allows one to avoid a drawback that fully automatic systems have: such as maps were produced to be read by human, it is impossible to reach 100% of automatic recognition. It leads that at the saturated images, the process of finding errors could be very complicated and can annualise all the automatization advantages. In our technology, combination of automatic and interactive techniques allows to perform object digitizing under operator control and correct data immediately if required.

Acknowledgment

This work was partially supported by INTAS 00-626 project.

References

- 1.Janssen R.D.T., Vossepoel A.M., Adaptive vectorization of line drawing images, *Computer Vision and Image Understanding*, vol. 65, no. 1, 1997, 38-56.
- 2.Katona E., Hudra Gy, An interpretation system for cadastral maps *Proceedings 10th International Conference on Image Analysis and Processing*, 792-797, 1999.
- 3.Kyong-Ho Lee, Sung-Bae Cho, Yoon-Chul Choy, Automated vectorization of cartographic maps by a knowledge-based system, *Eng. Appl. Artif. Intell.*, 165-78, vol.13, no.2, 2000.
- 4.Li L., Nagy G., Samal A., Seth S., Xu Y., Integrated text and line-art extraction from a topographic map, *Int. J. Doc. Anal. Recognit.*, 177-85, vol.2, no.4, 2000
- 5.Espelid R, Alvsvaag N A, Eileng J, Skauge I, Automatic digitizing of the color-layer of thematic maps, *Proc. IAPR Workshop on Machine Vision and Applications*, Tokyo, pp.299-302, 1990.
- 6.Ablameyko S, Bereishik V, Paramonova N, Marcelli A, Ishikawa S, Kato K, Vectorization and representation of large-size 2D line-drawings. *Journal of Visual Communication and Image Representation*, Vol.5, No.3, pp.245-254, 1994.
- 7.Agam G, Luo H, Dinstein I, Morphological approach for dashed lines detection, *Graphics Recognition*, R.Kasturi and K.Tombre Eds., *Lecture Notes in Computer Science*, 1072, pp.92-105, 1996.
- 8.Dori D, Wenyin L, Peleg M, How to win a dashed line detection contest, *Graphics Recognition*, R.Kasturi and K.Tombre Eds., *Lecture Notes in Computer Science*, 1072, pp.286-300, 1996.
- 9.Ablameyko S., Pridmore T., *Interpretation of line-drawing images*, Springer, 2000, 285p.