

Text/Graphics Separation and Recognition in Raster-scanned Color Cartographic Maps

Aurelio Velázquez¹ and Serguei Levachkine²

¹Program in Applied Math and Computing - Mexican Petroleum Institute (PIMAYC - IMP)

avelaz@imp.mx

²Geoprocessing Laboratory (GEOLAB)

Centre for Computing Research (CIC) - National Polytechnic Institute (IPN)

palych@cic.ipn.mx

Abstract. A method to separate and recognize the touching/overlapping alphanumeric characters is proposed. The characters are processed in raster-scanned color cartographic maps. The map is segmented first to extract all text strings including those that are touching other symbols, strokes and characters. Second, OCR-based recognition with Artificial Neural Networks (ANN) is applied to define the coordinates, size and orientation of alphanumeric character strings in each case presented in the map. Third, four straight lines or a number of “curves” computed as a function of primarily recognized by ANN characters are extrapolated to separate those symbols that are attached. Finally, the separated characters input into ANN again to be finally identified. Results showed high method’s rendering in the context of raster-to-vector conversion of color cartographic images.

1 Introduction

Cartographic color maps contain a lot of “punctual”, “linear” and area objects. To describe these objects there can be used symbols (portrays) and labels (alphanumeric characters) presenting a great variety of features, some of them in equal shape but in different color. Different colors are used to represent different objects, including a number of character’s fonts. These can be colored and following different paths in all kind of angles.

The development of a Geographical Information System (GIS) includes the selection of the paper and raster maps for vectorization Levachkine *et al.* [1]. To be included into GIS, the paper maps should be changed to a computer readable format, normally a raster format. After that, the raster maps can be converted into vector format that is most adequate to GIS. In the context of raster-to-vector conversion of graphical documents, the problem of text recognition is of special interest, because textual information can be used for verification of vectorization results (post-processing). The retrieval of all presented elements in a map can be made manually or supported by a computer system. In the former case, the map is scanned in a raster format and then converted to vector. Before a raster-to-vector conversion a map segmentation and recognition are usually employed Levachkine [25].

Some works on text processing in graphic images to be mentioned are as follows.

General frameworks. The text segmentation and its subsequent recognition in raster images are very difficult problems because, in general, there is either text embedded in graphic components, or text touching graphics Doermann [14]. These challenging problems have received numerous responses from the graphic recognition community Nagy [15]. However, there have not been developed efficient programs that solve the problem automatically. Thus, the main idea of the most works is to put the operator in the loop (even in the center of a computer system). As proposed, for example, by Ganesan [16], the operator can draw line through the text, marking it as text and revealing its orientation all in one step. Fletcher *et al.* [17] and Tan *et al.* [18] developed the algorithms to extract text strings from text/graphics images. Both methods however assume that the text does not touch or overlap with graphics. Wenyin *et al.* [7] recognized horizontal or vertical text. Luo *et al.* [3] detached characters from straight lines. For maps, the problem is much more complex since the touching or overlapping as well as many other character configurations are commonly presented in maps. Cao *et al.* [20] proposed a specific method of detecting and extracting characters that are touching graphics in raster-scanned color maps. It is based on observation that the constituent strokes of characters are usually short segments in comparison with those of graphics. It combines line continuation with the feature line width to decompose and reconstruct segments underlying the region of intersection. Experimental results showed that proposed method slightly improved the percentage of correctly detected text as well as the accuracy of character recognition with OCR.

Segmentation. Applying color and spatial attributes to segment thematic maps, Silva [2] used a 300-dpi resolution in a *RGB* color system to perform a Karhunen-Loeve transformation. Luo *et al.* [3] used the directional morphological operations. They coded images by run-length-encoded as an enchainned list, deleting the text that is represented by lines, and finally subtracting the new image from the original one to obtain an image with text without lines. In [4], Li described the Comb algorithm based on the best common structure of local minima defined at the moment to search for global minima. He used the concept of maximum a posteriori (MAP) and Markov random fields (MRF) as the frameworks. To segment text from engineering drawings Adam *et al.* [11] used Fourier-Mellin transform in a five-step process. Using a heuristics, they found broken chains. In [12], Hase *et al.* described a three-step algorithm of segmentation called "multi-stage relaxation". However, they do not recognize characters. In [5], Levachkine *et al.* used false colors in a *RGB* model. They applied different combinations of basic colors to segment map objects, and then a neighborhood analysis to recover or eliminate pixels.

Extraction and recognition. In [6], Myers *et al.* described the verification-based approach for automated text and feature extraction from raster-scanned maps. They used a gazetteer (toponym dictionary) to propose a forecasting hypothesis, which characters are in labels and where is their position in the map, having the information from another map in a different scale. Character and text boxes are used in [7] by Wenyin *et al.* The authors considered only horizontal and vertical text in which a character box is a rectangle with rate sides are no larger than 10 pixels to join character boxes. Thus, they built the text box that can grow horizontally or vertically under a threshold to fit the letters. Using directional morphological operations Luo *et*

al. [3] separated the text from lines but not from curves. Deseilligny *et al.* [8] proposed different knowledge levels to solve the problem. They begun with an analysis of related components (semiologic), then built the character chains (syntactic), detecting related characters (higher semiologic level) and, finally, following the natural language rules corrected the text (semantic). It seems too complex to be effective. Using templates Friscknecht *et al.* [9] linked them with symbols and characters. The approach does not require the complete template. It is pondered and hierarchically built. To retrieve street names Nagy *et al.* [10] used one of the four black layers. Taking the hue component from a *HSV* model for segmentation, they subtracted the street layer from the black layer and then made a connected component analysis to distinguish text characters. An efficient system to recognize characters by means of adaptive ANN is described in [13] by Velázquez *et al.* To train ANN, they used characters from a word processor in different fonts, sizes and inclinations by applying them to identify a great variety of characters in cartographic maps. This OCR algorithm is also used in the present paper. It is shown that using some “rules of thumb” we can obtain better results.

The rest of paper is organized as follows. In Section 2, we describe an alphanumeric segmentation-recognition system. In Section 3, we consider the touching and overlapping characters presented in raster-scanned color cartographic maps. A method (*V-lines* and *V-curves*) to separate and further recognize such characters is described in this section as well. Section 4 contains paper’s conclusion.

2 Segmentation of Overall Map Characters

A raster map has to be segmented first. All its elements should be retrieved with their coordinates and features, and then sent to the corresponding thematic layers. The layers can be symbols, landmarks, isolines, natural and artificial surroundings, words and numbers, lakes and other “punctual”, “linear” and polygonal bodies. Cartographic maps are the most complex graphic documents due to the high density of information that they contain. A typical example is shown in Figure 1a (*RGB* image).



Fig. 1. (a) Example of color cartographic map with different types of characters; (b) Gray-level image (*Y-luminance*) corresponding to the image of Figure 1

To obtain a binary image from color image, the former is usually changed to a gray-level. One way to make this change is to convert the RGB model to the YIQ model, where the luminance (Y) is a gray-level image. Another way is to average the R , G and B values. Figure 1b shows the gray-level image obtained from the image of Figure 1a. In this work, we used both conversion procedures as well as their combination as described in [5] by Levachkine *et al.* The binary image can be obtained applying a threshold to the gray-level image. Usually, the threshold is selected from the histogram computed by the frequency's value that shows the pixels from the gray-level image.

The selection of the threshold is the most critical task in a transformation of the image to binary format. Figures 2a and 2b illustrate this statement: 2a (a high-value threshold) and 2b (a low-value threshold).

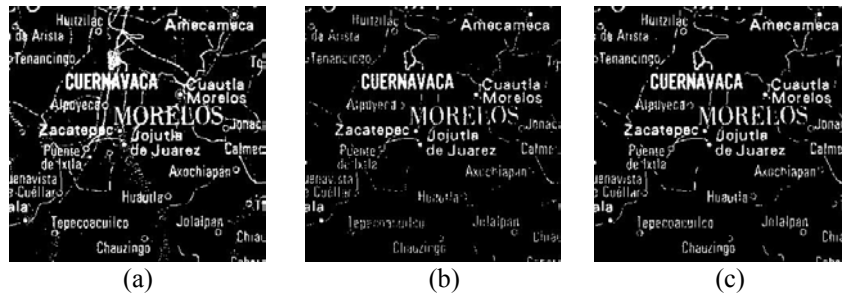


Fig. 2. (a) Binary image of Figure 1b (a high-value threshold); (b) Binary image of Figure 1b (a low-value threshold); (c) Binary image of Figure 1b (global dynamic thresholding [5])

In Figure 2a, some “red” lines are touching the characters. In Figure 2b there are no more “red” lines, but some characters are disappeared such those that belong to the word “Tepecoacuilco”.

In contrast, a method described in [5] uses false color combinations to obtain the binary image from a RGB image by applying a global dynamic thresholding. This method was primarily developed for the cartographic image subject domain. We obtained sufficiently good results in this domain [5]. Later, it was generalized to arbitrary image subject domain by Alexandrov *et al.* [21], making it application-independent. Figure 2c shows a binary image obtained with that method.

3 Separation and Recognition of Touching and Overlapping Characters

This is the main section of the paper. Based on detailed analysis of the textual characters presented in raster-scanned color cartographic maps and possible configurations of touching and/or overlapping text/graphics, we propose a simple method to separate the text from graphics component that is called V -lines. V -lines do work well in the case of characters with fixed inclination. However, they exhibit certain limita-

tions in the case of curvilinear text. To partially solve the problem of curvilinear text/graphics separation, we discuss a possible generalization of V-lines method, which we call *V-curves*.

To link all characters that belong to the same (and geographically meaningful) word, it is required to select all those that are related through the spatial relationships, the average size, color similarity, trend angle, etc. Gelbukh *et al.* [22][23]. All words that have been recognized, using the gazetteer, are sent to GIS, while those that were not identified, remained in the picture matrix.

3.1 V-lines (horizontal labels)

We have begun our analysis of cartographic textual characters of fixed inclination with a simple observation that such text can be accommodated between two of four (low case letters) rectilinear segments, called V-lines, while the upper case letters is considered apart as follows.

Capital letters nearly always have the same height, though some of them could be taller than the other letters that do belong to the same word such as the letter ‘Q’ shown in Figure 3.



Fig. 3. Capital letters forming a word, where the letter ‘Q’ is taller than the other letters

In this case, a recognition algorithm executed for this type of letters would have to recognize the characters touching or overlapping other objects. This sometimes reduces an efficiency of a recognition system. However, as a rule, this case is easily to maintain.

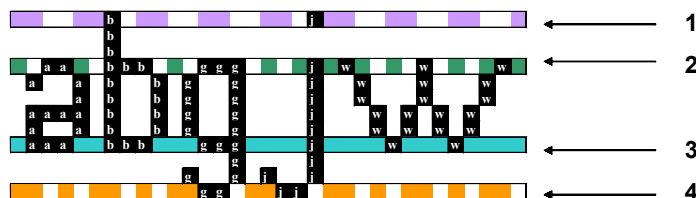


Fig. 4. Four rectilinear segments related to low case letters showing different combinations

On the other hand, low case letters are always related to two of the four imaginary lines shown in Figure 4 and ordered from the top to bottom: one, two, three and four. The possible combinations of two of these lines are “two-three”, “one-three”, “two-four”, and “one-four”.

To obtain the four V-lines, we employ the following steps:

- The imaginary lines are grouped under their position in the image matrix of map in accordance with the first and last matrix’s row of each character that belongs to the cluster,

- The corresponding rows are numbered from the top to bottom,
- The row with the most bottom character's occurrences (a three-pixel tolerance is admitted) is selected as line three and an imaginary line is traced,
- The row with the most top character's occurrences is selected as imaginary line two. Thus obtained imaginary lines are the basic V-lines,
- If there is another row down the row three such that the distance (in pixels) between both is smaller than one third of the distance between row two and row three, then it is labeled as the fourth line,
- A similar process is employed to obtain the line one, at the top, if exists.

There are characters that are touching other characters (or even other objects) with the same attributes. Thus, it is impossible to make a "natural" segmentation, because the characters share similar features. Two examples of these problems are shown in Figure 5a (the character 'S' is touching a symbol) and Figure 5b (the character 'S' is overlapped two other characters 'n' and 'o'). Figure 6a shows the corresponding binary representation of Figure 5a.



Fig. 5. (a) Character touching a symbol with similar features; (b) Character overlapping two characters with identical features

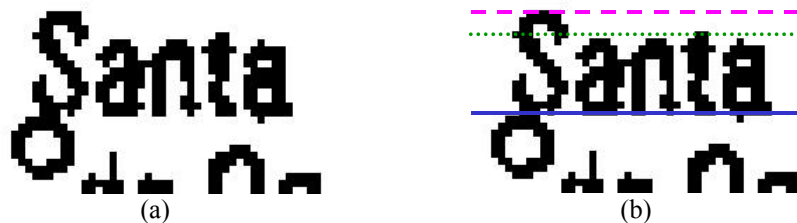


Fig. 6. (a) Binary image of Figure 5a; (b) V-lines to separate the letter 'S' from the symbol is below it in Figure 6a

Despite the letters 'a' and 'n' in Figure 6a are linked by a pixel, they could be moved away in the preprocessing step, under the 'neighborhood pixel analysis'. At this time, we obtain the chain "anta" identified by the set of ANN [13].

Extrapolating the lines shown in Figure 4, it is possible to separate the letter 'S', shown in Figure 6a from the symbol is below it, and then identified it with the set of ANN [13]. After this process is finished, it is possible to obtain the whole chain to form the word "Santa". The word is checked in a gazetteer to verify that it is a valid word for this geographical location [22][23]. Figure 6b shows the lines one, two and three used (line four is under the symbol touching 'S', but it is so far from line three).

3.2 Growing Rectangle (slanted text)

If the upper case characters are not horizontal, we can trace a “diagonal” rectangle following the same angle that those letters have. In Figure 7a, the first ‘S’ is touching the river’s line which name is ‘BALSAS’.

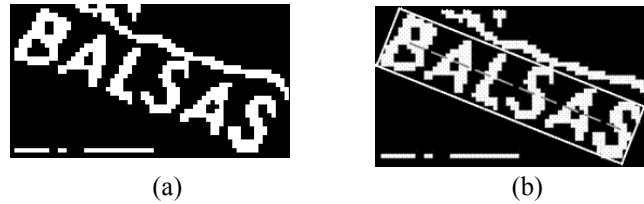


Fig. 7. (a) Label of ‘BALSAS’ river is touching the line of the river at the first letter ‘S’; (b) Starting with a line following the label angle, a dynamic rectangle is built

With the other five letters (‘B’, ‘A’, ‘L’, ‘A’, and ‘S’); we can build a rectangle that covers all the word letters. Using the upper and lower pixels from the first and last characters of the chain, we compute their mean value obtaining two points to trace an imaginary line with them as shown in Figure 7b. Then the left or upper (if the chain is vertical) point is moved one pixel up (or right) and one left (or down) following the left (or upper) line of the rectangle so that we have two new points, the right or lower point is moved one pixel down (or left) and one right (or up) following the right (or lower) line of the rectangle so that there are four points for the first dynamic rectangle. Computing the position of each character in the largest lines, we can find if there are outside pixels. The shortest lines are used for the same purpose with the first and last characters. If there are outside pixels in a line, it is moved one pixel in corresponding direction. If more than one line has outside pixels, all those lines are moved. The process is continued until no more outside pixel are found. Now, it is possible to identify the missing letter ‘S’ using only the pixels inside of the rectangle box belonging to letter ‘S’, sending them to ANN and testing the word ‘BALSAS’ in the gazetteer.

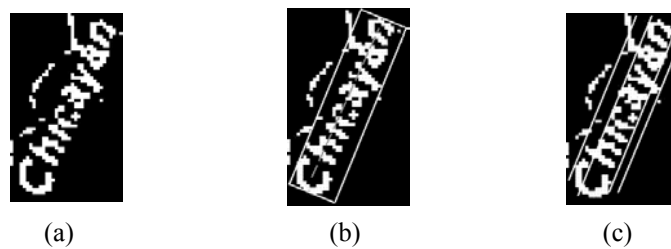


Fig. 8. (a) Label of ‘Chicayán’ river is touching the line of the river with the letter ‘á’; (b) First, it is necessary to build a rectangle as it was made with capital letters; (c) Four lines to unglue letter ‘a’ from the riverbed

On the other hand, for upper and lower case letters, four lines should be computed. Figure 8a shows the word ‘Chicayán’ touching a line of the river labeled for it. Using the procedure employed for capital letters, it is possible to construct a rectangle as

shown in Figure 8b. The largest lines will be used to find the four V-lines, if they are present. Normally, three of them are present. Each line should be moved with pixel-by-pixel procedure until it is reached the next numbered level. An additional adjustment is made: each ending point is moved up three pixels and the line that better fits one of the inner lines is selected as the “leader”. To the other three lines, if exist, we assign the same angle that the leader has. Figure 8c shows these lines.

Cutting the label with line one, the missing letter can be analyzed. It could be recognized as a letter ‘d’, but the word ‘*Chicaydn*’ does not exist in gazetteer. Thus, we attempt now with line two. Then, the letter can be interpreted as an ‘a’ and the word ‘*Chicayan*’ is already in the gazetteer and at right geographical location.

There are other manners in which objects can touch an alphanumeric character: at its left or right, at its top or bottom as in example shown in Figure 9a, where letter ‘M’ is touched at its left by a state boundary.

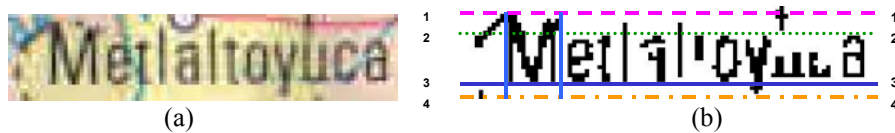


Fig. 9. (a) The word is touched at one extreme of the chain; (b) “Growing” rectangle to identify the letter ‘M’

The V-lines are not helpful themselves to separate the characters. However, the V-lines are useful to build a “growing” rectangle that is fitted to the character’s pixels, identifying the characters with the ANN and using the gazetteer until it matches to a correct word. The growing rectangle, around letter ‘M’ is shown in Figure 9b.

To build the rectangle, we employ the following steps:

- If the character is at left of string, we start at the beginning of the first letter. At this moment, we have lines two, three and four; there is another line, the first from the top to bottom. We use a tolerance of one third of the distance (in pixels) between lines two and three. A perpendicular line, that begins at line three plus tolerance (in pixels) and ends at line two minus tolerance (in pixels) is moved left until an appropriated pixel is found,
- The line moved is the first line of the rectangle. Other is formed by copying this line two pixels left. The other lines are two and three, unless there are pixels outside those lines, but inside of the tolerance. The first line found is the “anchor”, all others can be removed, and
- Left line is moved pixel-by-pixel. There could be a motion of upper or lower lines always inside of the tolerance. Each time the line is moved, the pixels in the rectangle are analyzed and tested by ANN, the gazetteer and the procedures described in [22][23]. The process is continued until a correct word is found or the distance between the line and the anchor is more than one and half times of the distance between upper and lower lines.

Unfortunately, there are some man-made errors on the maps and, even though our method outputs the complete chain of characters, the word can be misspelled. Figure 10a shows a word where letters ‘l’ and ‘n’ are touched by two lines. After the processing with V-lines shown in Figure 10b is applied the chain “*Tulacingo*” is built, but

this word is not in the gazetteer, because the right word is “*Tulancingo*” in which the letter ‘*n*’ was missing. This error can be corrected by another procedure [22][23].



Fig. 10. (a) Original chain touching two arcs of a boundary; (b) Characters were unglued and the chain ‘*Tulacingo*’ was build

3.3 More Difficult Cases

On the other hand, some labels are nearly impossible to detect because the background features are too close to their own features. Figure 11a shows such a label and Figure 11b shows its binary image with the damaged characters hard to identify.

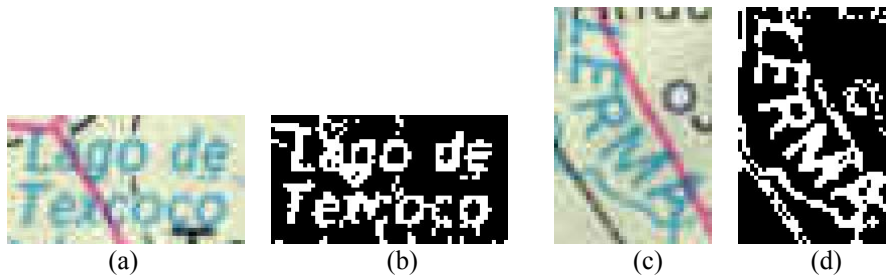


Fig. 11. (a) Label overlapped by other objects with similar attributes; (b) Binary image of Figure 11a, showing distorted characters; (c) Original image; (d) Binary representation of c

Another example is shown in Figure 11c. It is impossible to detect (in automatic way) the chain of characters because all of them are touching other elements.

In last two cases the operator intervention is certainly required as by Ganesan [16] or (better) by Gelbukh *et al.* [22][23]. In the former case the computer system provides useful hints to support the operator’s decisions. The operator’s task is to choose the best one for the given map situation and context. For example, in [23], a set of “name placement” strategies was proposed that aim to associate the name with its correct geographic location. As we mentioned above, the name placement is based on a number of the sources of evidence that included but not limited to, textual, spatial distribution, geographic, notational, and linguistic information.

3.4 V-curves (curvilinear labels)

Figure 12a displays a curvilinear text associated to a riverbed with the letter ‘g’ touching the line of the river (Figure 12b shows the corresponding binary image).



Fig. 12. (a) Color image with a curvilinear text “Tulancingo” (name of the river); (b) Binary image of Figure 12a

An application of V-lines method is difficult in this case. However, we can use the following procedure that we call *V-curves* to solve the problem. The text is divided into blocks of fixed (or nearly fixed) inclination by applying an algorithm similar to “growing rectangle” (Section 3.2). To each block the V-lines method is applied. Thus obtained lines are connected by a linear extrapolation, forming linear splines. These splines are V-curves. The following steps are similar to those that were used in V-lines method. To form meaningful words from disjoint text strings of each block, we use “spatial neighborhood semantic analysis” based on two sources of evidence as described in [22][23], Sections 5 and 6.

4 Conclusion

A method to separate and recognize touching and overlapping characters in raster-scanned color cartographic images has been presented. The algorithm performs the segmentation of the character layer and most valid (or “geographically meaningful”) words are built. This algorithm can detach nearly all characters touching other elements in any direction. Though some words cannot be obtained complete, the system is able to suggest one word from a gazetteer to support the operator decision to resolve the ambiguous cases. OCR-based recognition procedure with ANN called ASIMILA (a modification of the backpropagation paradigm) applied to the case of study possesses some peculiarities. ANN have been trained and tested first with synthetic characters. After that the same ANN have been employed for the characters of cartographic maps on a set of 2,125 samples. The results gave 93.21% of success [13]. These results have been improved with the V-lines/V-curves to some 96.73%. Even more efficient machine recognition of toponyms is reached by applying the spatial neighborhood semantic analysis in combination with the method herein described [22][23].

Acknowledgments

The authors of this paper wish to thank the Centre for Computing Research (CIC-IPN), General Coordination of Postgraduate Study and Research (CGEPI-IPN), National Polytechnic Institute (IPN), the Program in Applied Math and Computing of the Mexican Petroleum Institute (PIMAYC-IMP), the Mexican National Council for Science and Technology (CONACYT) and National Researcher Network (SNI) for their support.

References

1. Levachkine, S., Polchkov, E.: Integrated Technique for Automated Digitation of Raster Maps. *Revista Digital Universitaria (RDU)*. Vol. 1, No. 1 (2000). On-line: <http://www.revista.unam.mx/vol.1/art4/> (ISSN 1607-6079)
2. Silva, C.J.: Segmentation of Thematic Maps using Color and Spatial Attributes. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 221-230
3. Luo, H., Rangachar, K.: Improved Directional Morphological Operations for Separation of Characters from Maps/Graphics. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 35-47
4. Li, S.Z.: Toward Global Solution to Map Image Restoration and Segmentation using Common Structure of Local Minima. *Pattern Recognition*. Vol. 33, No. 1 (2000) 715-723
5. Levachkine, S., Velázquez, A., Alexandrov, V., Kharinov, M.: Semantic Analysis and Recognition of Raster-scanned Color Cartographic Images. *Lecture Notes in Computer Science*, Vol. 2390. Springer (2002) 178-189
6. Meyers G.K., Chen, C.-H.: Verification-based Approach for Automated Text and Feature Extraction from Raster-scanned Maps. *Lecture Notes in Computer Science*, Vol. 1072. Springer (1996) 190-203
7. Wenyin, L., Dori, D.: Genericity in Graphics Recognition Algorithms. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 9-20
8. Deseilligny, M.-P., Mariani, R., Labiche, J.: Topographic Maps Automatic Interpretation: Some Proposed Strategies. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 175-193
9. Frischknecht, S., Kanani, E.: Automatic Interpretation of Scanned Topographic Maps: A Raster-based Approach. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 207-220
10. Nagy, G.A., Seth Samal, S., Fisher, T., Guthmann, E., Kalafala, K., Li, L., Sarkar, P., Sivasubramanian, S., Xu, Y.: A Prototype for Adaptive Association of Street Names with Streets Maps. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 302-313
11. Adam, S., Ogier, J.-M., Cariou, C., Mullot, R., Labiche, J., Gardes, J.: Symbol and Character Recognition: Application to Engineering Drawings. *International Journal on Document Analysis and Recognition (IJ DAR)*. No. 1 (2000) 89-101
12. Hase, H., Shinokawa, T., Yoneda, M., Suen, C. Y.: Character String Extraction from Color Documents. *Pattern Recognition*, Vol. 34, No. 1 (2001) 1349-1365
13. Velázquez, A., Sossa, J. H., Levachkine, S.: Reconocimiento Eficiente de Caracteres Alfanuméricos Provenientes de Mapas Ráster por Medio de Clasificadores Neuronales. *Computación y Sistemas*, Vol. 6, No. 1 (2002) 38-50 (ISSN 1405-5546)
14. Doermann, D.S.: An Introduction to Vectorization and Segmentation. *Lecture Notes in Computer Science*, Vol. 1389. Springer (1998) 1-8

15. Nagy, G.: Twenty Years of Document Image Analysis in PAMI. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1 (2000) 38-62
16. Ganesan, A.: Integration of Surveying and Cadastral GIS: From Field-to-fabric & Land Records-to-fabric, Proc. 22nd ESRI User Conference, 7-12 July 2002, Redlands, CA, USA (2002) on-line: <http://gis.esri.com/library/userconf/proc02/abstracts/a0868.html>
17. Fletcher, L.A., Kasturi, R.: A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 6 (1988) 910-918
18. Tan, C.L., Ng, P.O.: Text Extraction using Pyramid. Pattern Recognition, Vol. 31, No. 1 (1998) 63-72
19. Velázquez, A.: Localización, Recuperación e Identificación de la Capa de Caracteres Contendida en los Planos Cartográficos. *Ph.D. Thesis*. Centre for Computing Research-IPN. Mexico City, Mexico (2002) (in Spanish)
20. Cao R., Tam, C.L.: Text/Graphics Separation in Maps. Lecture Notes in Computer Science, Vol. 2390. Springer (2002) 168-177
21. Alexandrov, V., Kharinov, M., Velázquez, A. Levachkine, S.: Object-oriented Color Image Segmentation. Proc. IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2002), June 25-28 2002, Crete, Greece (2002) 493-498
22. Gelbukh, A., Han SangYong, Levachkine, S.: Combining Sources of Evidence to Resolve Ambiguities in Toponym Recognition in Cartographic Maps. In: Levachkine, S., Serra, J., Egenhofer, M. (eds.) Proc. 2nd Int. Workshop on Semantic Processing of Spatial Data (GEOPRO 2003), November 4-5 2003, Mexico City, Mexico (2003) 42-51 (ISBN 970-36-0103-0)
23. Gelbukh, A., Levachkine, S.: Resolving Ambiguities in Toponym Recognition in Raster-scanned Cartographic Maps. In: Lladós, J. (ed.): Proc. 5th IAPR International Workshop on Graphics Recognition (GREC 2003), July 30-31, 2003, Barcelona, Catalonia, Spain (2003) 104-112
24. Velázquez, A., Levachkine, S.: Text/Graphics Separation and Recognition in Raster-scanned Color Cartographic Maps. In: Lladós, J. (ed.): Proc. 5th IAPR International Workshop on Graphics Recognition (GREC 2003), July 30-31, 2003, Barcelona, Catalonia, Spain (2003) 92-103
25. Levachkine, S.: Raster to Vector Conversion of Color Cartographic Maps for Analytical GIS. In: Lladós, J. (ed.): Proc. 5th IAPR International Workshop on Graphics Recognition (GREC 2003) July 30-31, 2003, Barcelona, Catalonia, Spain (2003) 77-91