

Abdelhafid Bessaid¹, Hassane Bechar¹, M. Karim Fellah²

¹*Laboratoire de Génie biomédical, Département d'Electronique, Faculté des sciences de l'Ingénieur, Université de Tlemcen*

²*Faculté des sciences de l'Ingénieur, Université de Sidi Bel-Abbes.*

e-mail: a.bessaid@freesurf.fr

Image analysis and pattern recognition as tools in map interpretation

Received 11.06.2003, published 19.08.2003

In this article we propose a methodology, which uses image analysis and pattern recognition techniques for the acquisition and the interpretation of cartographic documents. The main subject on map interpretation is to make easy their exploitation by proposing some steps allowing the transformation of the maps in paper form to a numerical storage of information in computers. Our approach is aiming to construct a system that can generate a symbolic description of the map. This approach includes color segregation, buildings localization, symbols identification, roads extraction and related character string recognition. All the information must be encoded. A color scanner generates a multicolor raster image of the map. This image is split into layers of predefined map colors. Each layer is then vectorized. To signify the feasibility of the proposed system, a brief description of the system is discussed and some experimental results are shown.

1. INTRODUCTION

Recently, automatic map reading and interpretation has become an important topic in document analysis researches. Many methods are used, and the main subject on map interpretation [1, 10, 24] is to make easy their exploitation by proposing some steps allowing the transformation of the maps in paper form to a numerical storage of information in computers [1, 16, 19]. The subject on map recognition can be mainly divided into several research domains: road extraction, building localization, symbol identification and character string recognition. Many researchers have proposed systems to solve automatic map interpretation problems [4, 5, 7]. Many works have been published on the specific domain like cartographic document [2, 4, 10, 11], cadastral map [7, 8, 9, 17] or mechanical and electrical schemes [6]. In the past ten years, the cartographic domain has become more important for storing and organizing spatial information in computers. The large number of paper documents and the thousands of changes to be done each year for their updating make very necessary the use of computer for solving this task type. The key step to elaborate numerical map is the data acquisition. The result obtained by scanning operation requires a great space memory for the image storing and the only interest is the simple visualization. Indeed, it is necessary to attach attributes to the graphic data, there begin the long work of vectorization

[21, 23]. For our application, most of the maps are available only in paper form. The manual digitization requires considerable time and cost because of the large number of manual operations. It is then interesting to automate the process of extraction of the restrained information in maps to study by scanning and applying image analysis methods.

In this article, we propose a complete processing chain allowing the automatic vectorization of topographic maps. The first step of this automatic processing is the digitization of the document by scanner. This operation allows passing from a paper form document to a pixel form one which can be exploitable by computer. The second step includes all images analysis techniques [3] which transform the “pixel” document to high level information: regions identification, contour of buildings, symbols and character strings extraction. Beyond the gain of memory space, this information type allows extraction of only useful information from the map under study: roads, built regions, water... Each information stored can be then selected. It is also easy to modify numerically this information. Several systems have been developed for solving the task of automatic data extraction from the map. These systems require, as input data, separated map layers [5, 10] or map of one color [7, 8]. We used topographic map on a scale of 1:10.000 that are printed with four colors printing technique. The printing colors are brown, green, blue and black. Therefore, we developed appropriate methods for raster data processing and data organization. The main ideas of raster data processing are scanning with a red, green, blue (RGB) scanner, separation of color layers, raster-symbol and raster-object recognition, and vectorization. Results of these last operations are stored in the files text characterized by a low space memory. The paper describe the details of the vectorization system, it is divided into sections which give the major technical aspects. In section 2, we present the methodology used for the contour vectorization of built regions and line vectorization of character strings. Road extraction is another important task in map interpretation. Morphological operations are widely used to find the roads [11]. Section 3 describes the operation of colors separation. The methods of processing, image analysis, vectorization and the results obtained by the analysis typical map scene, are discussed in sections 4 and 5.

2. INTERPRETATION METHODOLOGY

The problem of interpretation [7, 8, 11] is a complex one for which it agrees to choose the best strategy. The topographic maps we use are printed in a four colors printing technique. A raster image of the map to be processed is created with a color scanner. Theses maps include road drawing, houses, and all buildings position, so they are frequently updated by specialised agencies. Maps on which are based our study are on a scale 1:10000 (1 cm paper = 100 m land). This scale is, indeed, the smallest not distorting the topographic reality. The proposed methodology is illustrated by Fig. 1.

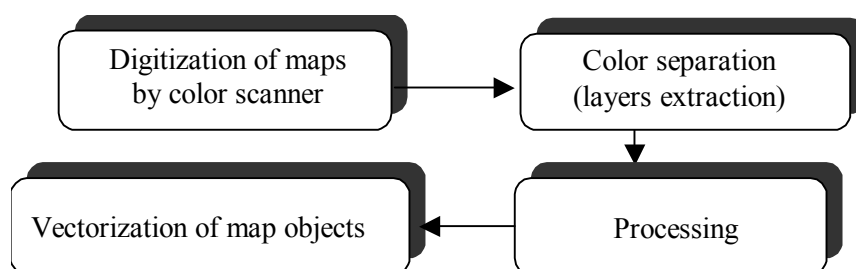


Figure 1. Interpretation methodology

Image acquisition is done by a color scanner with a resolution of 300 dpi. A symbolic description of all elements of the map (roads, houses, water, symbols...) is achieved in several stages. Indeed, the first, of primordial importance, separate the elements of the map according to their color. One gets layers containing information printed with the same color thus (information in the same way type). In a second phase, the extraction and the separation of layers representing elements corresponding to buildings and roads, then elements corresponding water. Note that the vectorization of regions is achieved after the extraction of contour whereas lines (roads, course of water) will be thinned first (skeleton) then vectorized [2, 6, 21]. In the last stage, we interest to the coding of notations and the recognition of symbols.

3. COLOR SEGMENTATION

3.1 Principle

The separation of color from a topographic map is a problem of image color segmentation [4]. Information, which we can extract from this type of map, is represented with the following colors: brown (roads and buildings), black (notations, the local limits and symbols), blue (courses of water, streams, rivers, castles of water, wells,...), green (cultivated zones, fields, vegetations). The dissociation of these four colors is a considerable simplification before the vectorization.

Images that we have in entrance are coded on three plans: red, green and blue. The color of each pixel is given by a vector with three components (i.e. RGB space color). The three elements of each vector corresponding to the RGB components of every pixel of the image to process are presented to the entrance of a classifier [12] in the goal to do a separation color. An example of data file with the three components R, G, B is given below (Table 1). The use of the color in image segmentation is a relatively recent research topic. Although one finds several algorithms [2, 4, 11, 14] of color segmentation, but the literature is not rich enough that for images in grey level. The method that we propose to solve the problem of the classification color uses the Kohonen's model [13] that we present in the following lines.

Table 1. Example of data file with the three components R, G, B

R	92	112	215	171	60	166	226	0	...
G	78	86	167	197	24	193	182	48	...
B	56	64	106	123	255	111	109	242	...
Color	black	black	brown	green	blue	green	brown	blue	...

3.2 Kohonen's model

- This model corresponds to a layer of neurons and one entrance layer (Fig. 2).
- The layer of entrance serves only to the presentation of entrance vectors (components R, G, B of the different pixels).
- An adaptation layer formed by a neuron network. These neurons are some simple linear and are connected to all components R, G, B of the entrance layer.

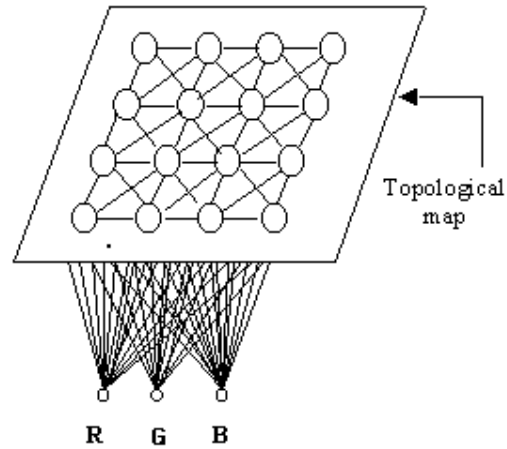


Figure 2. Topological map

Every neuron j of topological map calculates a distance between the x example presented to the entrance and its weight vector W_j (entrance vectors x and weight vectors W of neurons of the map have the same dimension). The neuron j^* (winner) is then the one that have the distance minimum:

$$\|x - W_{j^*}\| = \min \|x - W_j\|, \quad (1)$$

The Euclidean distance [12], very used in the domain of the classification, is given by:

$$d_j = \sum_{i=0}^N (x_i - W_{ij})^2, \quad (2)$$

where N is the dimension of the entrance vector, it is equal to 3 (RGB components) in our case.

Therefore, we determine the neuron whose weight vector is nearest, to the sense of the Euclidean distance, of the presented vector x . It adjusts its weights then in order to come closer more of the example presented again. In its topological version, every winning neuron incites these neighbors to also modify their weight in the same sense. If one notes j^* the neuron winning, V_{j^*} its neighbourhood and W_j the vector of weight of a j neuron, modifications that are going to take place after presentation of the vector of x entrance to the iteration at time " t " are going to be:

$$W_j^{(t)} = W_j^{(t-1)} + \alpha^{(t)} (x^{(t-1)} - W_j^{(t-1)}) \text{ if } j \in V_{j^*},$$

$$W_j^{(t)} = W_j^{(t-1)} \text{ otherwise,}$$

where $\alpha^{(t)}$ is a gain term ($0 < \alpha^{(t)} < 1$) that decreases in time.

Algorithm [12]:

Step 1. Initialize Weights

Initialize Weights from N inputs to M output-nodes shown in Fig. 2 to small random values.

Step 2. Present New Input

Step 3. Compute Distance to all Nodes

Compute distances d_j between the input and each output node j using

$$d_j = \sum_{i=0}^{N-1} (x_i^{(t)} - W_{ij}^{(t)})^2,$$

where $x_i^{(t)}$ is the input to node i at time t and W_{ij} is the weight from input node i to output node j at time t .

Step 4. Select output Node With Minimum Distance

Select node j^* as that output node with minimum distance d_{j^*} .

Step 5. Update Weights to Node j^* and Neighbors

Weights are updated for node j^* and all nodes in the neighborhood defined by $j \in V_{j^*}^{(t)}$

New weights are

$$W_{ij}^{(t)} = W_{ij}^{(t-1)} + \alpha^{(t)} (x_i^{(t-1)} - W_{ij}^{(t-1)}) \text{ for } j \in V_{j^*}^{(t)} \text{ and } 0 \leq i \leq N-1.$$

Step 6. Repeat by Going to Step 2

The data file dedicated to the training includes three columns of the RGB components and a fourth column in which we gave like label the name of the color corresponding to the R, G, B combination (Table 1). Note that a preliminary manual operation must be done by an operator where he must select on maps to study zones corresponding to these four colors (brown, black, green and blue) in order to construct a data file prototype. This data file will be used in the training phase of the network. Once the classification operation of pixels achieved, we get the four separated layers. This one corresponds to the brown layer (brown) for constructions, the blue layer for water, the green layer for vegetation and the black layer including the different notations. We proceed to the binarization of these four images obtained from the color segmentation (Fig. 3) in order to begin the phase of vectorization.

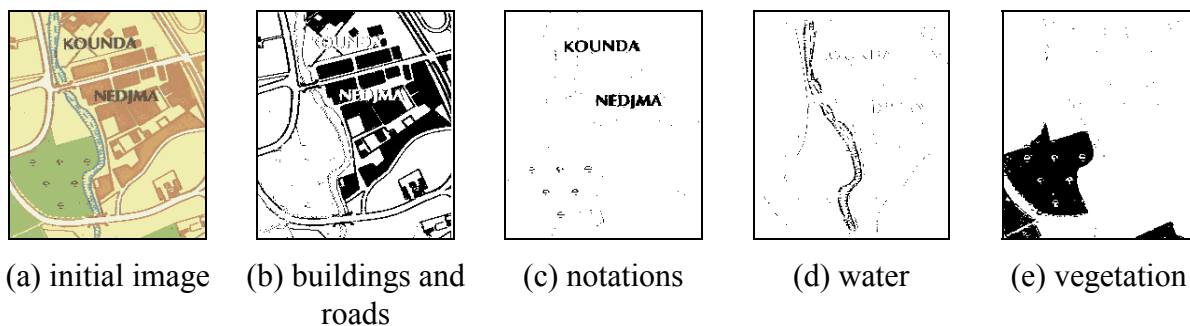


Figure 3. Operation of color separation

4. FILTERING

Files obtained after digitization of image by scanner include nuances not existing initially. These create parasitic pixels having identical colors to useful elements of image (Fig. 3), it complicated the operation of color separation since some parasitic pixels also appear in images obtained after separation. A manner to eliminate them is to apply a pass-low filtering that is going to clean these high frequency noises. The median filter [3], very known for its cleaning properties, is very well adapted to this situation and possess the advantage to keep intact contours of elements of image, but its inconvenience resides in the fact that it must be applied several times in succession to get good results. Fig. 4(a) shows effects on image symbols after the parasitic pixel suppression by filtering using median filter. Indeed we notice a deterioration of characters. It is owed to the suppression of pixels being on the side of characters and is surrounded of a certain number of white pixels. They are isolated, they are suppressed therefore.

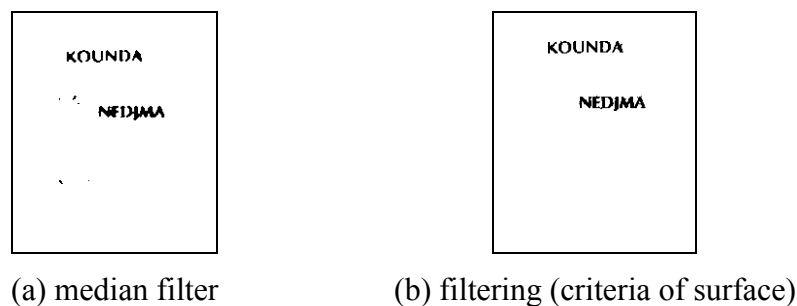


Figure 4. Filtering of notations

Another manner to eliminate these noises is to use a filtering following criteria of surface. One notes that the parasitic pixels are isolated either in some pixel group. One calculates the surface of all pixel groups then one suppresses those whose surface is lower to a threshold.

For a map on scale 1:10.000 and scanning with a resolution of 300 dpi (12 pixels by mm), the threshold is fixed to 10 pixels. A fast count allows to adapt the threshold to all different profiles. The result of this filtering type is given by Fig. 4(b).

5. VECTORIZATION

The vectorization is an important stage in the process of cartographic document interpretation, it allows to pass from a “raster” image toward a vectorial representation. The factor of reduction between the initial image and the vector image is very important, what is very interesting for industrial. Several methods of vectorization exist [6, 10, 21], we distinguish two classes of algorithms: the vectorization of skeleton and the vectorization of contour. The principle of the vectorization is the representation by a whole segments stocked instead of a whole of points. Our approach consists in storing some directions from a start point with the manner that can minimize size of the file containing the useful information. The vectorization is done on elements having a width of 1 pixel. For the full regions (buildings, cultivated zones,...), this operation is achieved after the extraction of contour of

these regions. For elements curves (roads) and lines of transition [15] (course of water, lakes, streams,...) that are represented by lines of some pixels of thickness and for notations (characters) considered as related regions, we do a set of thinning operations to get a skeleton [3, 20] of which the algorithm, compared to the one of the contour extraction, is less fast, but provides some satisfactory results.

5.1 Extraction of buildings

The built zones and roads are represented by Fig. 3(b) with the same color (brown). We can recognize buildings easily, it is some full graphic entities of big thickness with regard to the other elements of the map. To eliminate the undesirable elements (roads that have a thickness of N pixels) and to keep only the useful elements (buildings), image will be eroded N times therefore ($N=2$ in Fig. 5(a)). The erosion is a destructive operation, therefore, one can not restore the destroyed pixels. Nevertheless, one tempts to recover a part of these pixels by application of a dilation (Fig. 5(b)). Some regions representing buildings present shortcomings (holes) resulting of the notation superposition (characters). These shortcomings are corrected with the help of growing region techniques using a certain number of rules [4] for example: change the value 0 by the value 1 of pixels belonging to a closed region and formed of value 0 pixels and that in one of the other layers corresponding to the black, green or blue color these pixels possess the value 1.

5.2 Chaining

The stage of chaining consists in joining the related points between them to form a list of points. For the vectorization of buildings, it is not necessary to stock all full graphic entity pixels, only those belonging to the contour will be taken into account. After extraction and follow-up of contour, a chain coding or Freeman coding [3] is done in order to store the direction until the neighboring pixel and that, for every pixel of the contour.

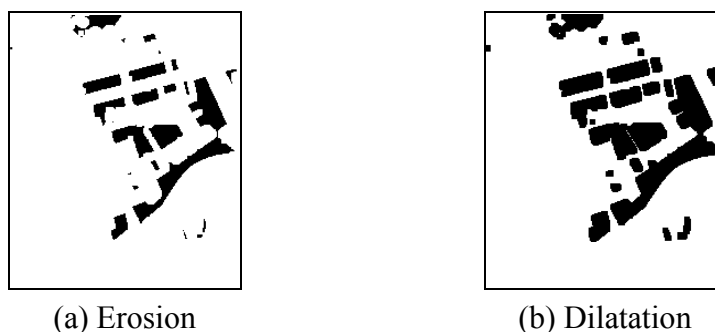


Figure 5. Morphological operations

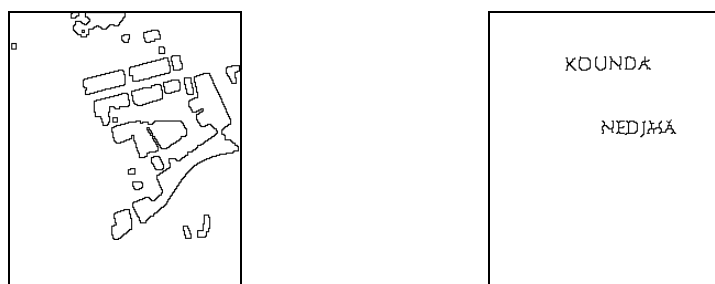


Figure 6. Contour of buildings

Figure 7. Skeleton

The procedure of coding requires a start pixel and browses the contour (high toward low and left toward right of image) until there are not any more neighbors or recover the start pixel. Fig. 6 represent the contour of buildings corresponding to the initial image of Fig. 3(a). Buildings have in general relatively straight contour. To optimize the gain of space memory, one can adapt run-length encoding to the coding of chain. Indeed, what characterizes a segment, it is the long succession of the same value of direction. This coding includes a number of identical directions followed, every time, by the value of the direction. Results of the vectorization are stored in a text file. This one contains results with a very explicit form and has the same name than image to analyze but with extension txt. The first four lines of this file are for the general information: number of lines, number of columns, resolution and scale of the map. A textual description of the element to vectorized is also presented. The following concerns coordinates of the first point of every entity, the dimension of the vector (number of directions) and the value of Freeman directions. These files results possess some very lower sizes to those of initial image files, we noted a compression of 90%.

5.3 Coding of notations

The use of notations in maps is indispensable since it allows to identify objects as roads, rivers and the different constructions or to name regions. Notations represented on maps distinguish themselves by their sizes and are generally aligned but can present very variable orientations. The main difficulty that present the process of textual data processing with regard to the classic systems of Optical Character Recognition (OCR) holds the fact that information is multi-oriented and multi-scale [22]. After the processing of zones built, the remaining objects to extract concern the textual zones therefore [25]. We present in this part the extraction and the coding of notations. The methodology of coding proposed is simple, in the first phase we get image with only notations. This can be obtained with the separation operation of black color. Image is then filtered in order to eliminate the parasitic pixels. After binarization of image, a thinning operation of image elements is done in order to store only the fine entities (skeleton) of notations having a thickness of one pixel. The last stage concerns the vectorization of notations, once again, the chain coding [3] has been chosen. This one guarantees a storage sure of every entity. Fig. 3(a) shows a piece of a map obtained by color scanner. Image gotten by extraction of black color elements includes some unacceptable parasitic pixels for a good use of image (Fig. 3(c)). This one is filtered then according to the criteria of surface presented in section 4, the result of this filtering is shown by Fig. 4(b) and Fig. 7 is the result of the thinning operation of elements (characters skeleton) of the map. Results of the notation coding are also stored in a file text. For the reconstitution of elements vectorized, one represents buildings by their contour and notations (characters) by their skeleton.

6. CONCLUSION

We proposed a general method for the vectorization of elements of a map, this one has been applied first to buildings and the coding of notations. Results that we present in this article are obtained with low-level tools of vectorization. They are validated on a meaningful number of maps. Note that the vectorization method of the strong elements (buildings) can not be efficient that while intervening on an image not including elements curves (roads,...); these last must be camouflaged or vectorized separately. The choice of the number of iterations, during erosion and dilation operations, is let to the user's latitude; indeed, this parameter essentially depends on the resolution of scanned maps, it would be necessary to adjust it consequently therefore.

For the coding of notations, and although it is consumer of time, we opted for the method of skeleton, seen its interesting precision. Another remark imposes itself, seen the small size of scanners, or the constraint of the space memory obliges us to break up maps. The vectorization should hold amount of the different fragments of maps by the use of methods for the automatic compilation of mosaics.

This work is a contribution to the vectorization and the cartographic document interpretation. It must however be refined in order to improve capacities of the method by providing of new tools low-level coming to complete already those available. Our present developments aim to conceive other tools allowing the vectorization of the elements curve (roads, freeways,...) and lines of transitions (course of water, rivers,...).

Another as important phase is the recognition of characters that represents an important domain in systems of cartographic document analysis. The particular textual data character incites us to search for methods that permit to hold amount of the specificity information on the multi-oriented and multi-scale characters.

REFERENCES

- [1] M. P. Deseilligny, R. Mariani, J. Labiche, R. Mullot. Topographic maps automatic interpretation: some proposed strategies. *Lecture Notes in Computer Science*, vol. 70, pp. 438–451, 1998.
- [2] A. Bessaid, M. K. Fellah, S. Mendil et M. Merrakchi. Vectorisation de cartes scannées. 1st National workshop on signal & processing NWSIP'98. Sidi Bel Abbès, pp. 28–31, 1998.
- [3] J. P. Cocquerez et S. Philipp. *Analyse d'images: filtrage et segmentation*. Masson, 1995.
- [4] N. Ebi, B. Lauterbach, and W. Anheier. An image analysis system for automatic data acquisition from colored scanned maps. *Machine Vision and Application*, pp. 148–164, 1994.
- [5] R. Mariani, F. Lecourt, M. P. Deseilligny, J. Labiche, Y. Lecourtier. *Interprétation de cartes géographiques. Algorithme de reconstruction des réseaux hydrographiques et routiers. Traitement du signal*, 1995.
- [6] Pascal Mercy, Brunot Taconet, J. Luc Maini. Une méthode de vectorisation de dessin techniques adaptée aux plans mécaniques. 16^{ième} colloque GRETSI, 15–19 Septembre, 1997, Grenoble, pp. 869–872.
- [7] J. M. Ogier. *Contribution à l'analyse automatique de documents cartographiques: interprétation de données cadastrales*. Thèse de Doctorat. Université de Rouen, 1994.

- [8] Ogier J. M, Mullot R, Labiche J, Lecourtier Y. Interprétation de document par cycles «perceptifs» de construction d'objets cohérents. Application aux données cadastrales. *Traitem. du signal*, vol 12, №6, pp. 627–637, 1995.
- [9] Christophe Pieplu, Ali Al-Hamdi, Rémy Mullot, Jean Marc Ogier, Patrick Dumas. Système d'interprétation de documents techniques. 16^{ième} colloque GRETSI, 15–19 Septembre, 1997, Grenoble.
- [10] Mullot R. Interprétation de documents techniques et cartographiques: Algorithmes et systèmes. HDR, janvier 2000, Université de Rouen.
- [11] Chung Lin Huang and Wei Lin. Automatic taiwanese municipal color reading system. *Vision Interface'99*, Trois rivières Canada, 19–21 May.
- [12] Richard P. Lippman. An introduction to computing with neural nets. *IEEE ASSP Magazine*, April 1987.
- [13] T. Kohonen. SOM_PAK, The self organizing map program package, November 1992, Helsinki University of Technology.
- [14] Olivier Lezoray et al. Segmentation d'image couleur: Application en microscopie cellulaire. *Traitement du signal 2000*, Volume 17, №1.
- [15] R. Kasturi et al. A System for Interpretation of Line Drawing. *IEEE Trans. on PAMI.*, 12(10), 1990.
- [16] B. Pasternak, The Role of Taxonomy in Drawing Interpretation. *Proc. ICDAR'95*, Montreal, Canada, pp. 799–802, 1995.
- [17] J. M. Ogier, R. Mullot, J. Labiche, Y. Lecourtier. The Semantic Coherency: The Basis of an Image Interpretation Device – Application to the French Cadastral Map Interpretation. *IEEE Trans. on System, Man and Cybernetics (SMC): Part B – Cybernetics*, 30(2), pp. 322–338, April, 2000.
- [18] L. Boatto and al. An interpretation system for land register maps. *IEEE Computer Magazine*, 25, pp. 25–33, 1992.
- [19] S. Suzuki, T. Yamada, MARIS: Map Recognition Input System. *Pattern Recognition*, 23, pp. 919–933, 1990
- [20] L. Lam, S. W. Lee, C. Y. Suen. Thinning methodologies – a comprehensive survey. *IEEE Trans. on Pami*, 14(9), pp. 869–885, 1992
- [21] R. M Haralick. Performance Characterization in Image Analysis: Thinning a Case in Point. *Pattern Recognition Letters*, 13(1), pp. 5–12, 1992.
- [22] M. P. Deseilligny, H. Le Men, G. Stamon. Character string recognition on maps, a rotation-invariant recognition method. *Pattern Recognition Letters*, 16, pp. 1297–1310, 1995
- [23] O. Hori, D. Doermann. Quantitative measurement of the performances of raster-to-vector conversion. *Lecture Notes in Computer Science*, vol. 1072, pp. 57–68, 1996.
- [24] F. Dupont, M. P. Deseilligny et M. Gondran. Automatic interpretation of scanned maps: reconstruction of contour lines. *Lecture Notes in Computer Science*, vol. 1389, 1998.
- [25] K. Tombre, A. K. Chhabra. Graphics recognition – algorithms and systems. *Lecture Notes in Computer Science*, vol. 1389, 1998.