Toponym Recognition in Scanned Color Topographic Maps

Joachim Pouderoux, Jean-Christophe Gonzato, Aurélien Pereira, Pascal Guitton

Toponym Recognition in Scanned Color Topographic Maps

Joachim Pouderoux Jean-Christophe Gonzato Aurélien Pereira Pascal Guitton
Iparla Project (LaBRI - INRIA)
University of Bordeaux, France
firstname.name@LaBRI.fr

Abstract

Topographic paper maps are a common support for geographical information. In the field of document analysis of this kind of support, this paper proposes an automatic approach to extract and recognize toponyms. We present a technique based on image segmentation and connected component processing. Different filtering stages ensure the consistency of plausible characters and strings. Detected text areas are used to feed an OCR software and the recognized words are analyzed and corrected. The main advantage of our technique is that no assumption is made about the character font, size or orientation. Experimental results obtained are encouraging in term of recognition efficiency.

1 Introduction

With the evolution of data acquisition, storage techniques and the growth of planet-wide networks accessible through a large amount of different devices, Geographical Information Systems (GIS)-based applications are more and more used and Global Positioning Systems (GPS) and route suggestions systems are widespread.

While modern *topographic maps* (ie. maps describing the topology of a part of the earth) also known as *contour maps* are issued from GIS, large libraries of older paper maps are still available and used in many applications: historical researches, urbanism, etc. The digital acquisition of these maps becomes necessary to analyze and use the represented terrains with today's tools.

Contrary to what we could think, feature extraction from topographic maps is a very difficult problem as explained in [8], mainly because of the acquisition noise and the poor accuracy of some features. It is also true for text extraction especially because text is often embedded in graphic components or text touches graphics [3]. For instance, a place name can be intersected by a road drawn using the same color.

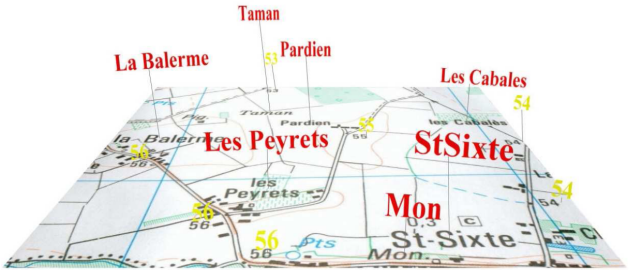

Figure 1. Toponyms extracted from a color topographic map.

In this paper we focus on the extraction of labels. Labels are words (ie. strings of characters) which constitute the textual layer of the map. In topographic maps, most of the labels are *toponyms* (ie. names of places or regions) but one could find some other textual information such as elevation values.

Given a scanned topographic map, the aim of our algorithm is to generate a text file containing recognized words and their coordinates on the original map using a standard (freeware or commercial) OCR software. The main goal of our method is to generate a set of images containing potential strings by detecting them on the map and to filter OCR results in order to obtain a consistent vectorized toponyms data file. Figure 1 shows a 3D view of a topographic map where extracted and recognized toponyms float on top.

This paper is organized as follows: in a first part we analyze previous approaches that have been proposed for the general problem of text extraction and recognition from paper based documents. In section 3 we present the 4 steps of our technique. Finally we present and discuss the obtained results before to conclude.

2 Related work

Text detection and recognition are very difficult problems which have generated a lot of research work. The

main difficulty lies in the graphics that surround or intersect the text. Hence, most of the techniques described in the literature are very specific in respect to the kind of documents they can process. The main feature of a good text detection algorithm is to be independent in respect to text font, size and orientation. One can distinguish two main families of methods: (i) methods devoted to detect text in paper-based documents, (ii) methods developed for text detection in videos or textured images. The later methods are mostly based on texture analysis (for example, refer to [12, 7, 6, 14]) and so are not very efficient on maps on which numerous features with similar properties than text are present.

An early approach in [5] introduced an algorithm which permits to extract text on binary images such as electronic circuits. Using connected components, a selection of possible characters is performed using the components' characteristics. The characters are merged into words using Hough Transform. This method, which inspired our paper, gives good results and has the advantage to be usable for various orientations and sizes of texts. Unfortunately, it does not support the overlapping of text and graphics elements of images which commonly encountered on topographic maps.

A method for reconstruction of characters string adapted to cartographic context has been presented in [13]. Connected components are analyzed and recognized in a set of possible fonts, then, strings reconstruction is reduced to a graph optimization problem. This is a high level method, however, it supposes a knowledge of fonts appearing on the maps and a segmented binary map while the paper does not address the problem of color map segmentation.

In [11], authors develop a model to extract black characters in urban maps. They use street lines data to recover the characters composing street names. Due to the possible overlapping between street names and street lines describing the street, they develop a specific OCR algorithm to produce an efficient text file of street names. [2] proposes a similar model but include a specific treatment of character extraction on text/graphic overlapped zones.

In [4], the first step of the algorithm is based on the detection of possible text areas. The characters are then interpreted as high level horizontal variations of gray scale and are extracted by a specific filter based on horizontal gradient computation. Unfortunately, this method is limited to horizontal text detection.

More recently, a text/graphic separation method applied to color thematic maps has been described in [10, 9, 15]. The authors describe a segmentation technique based on 24 images obtained by a combination of color components (R, G, B, (R+B)/2, etc.). The final binary image resulting from the mix of theses 24 images is then used in an OCR-based recognition system with neural networks. Segmentation results were not very convincing with the topographic maps we own and the memory consumption can be a real limitation when processing large images.

3 Toponym extraction and recognition

3.1 Overview

Given a color raster map, the aim of our method is to produce automatically a vectorized representation of the textual layer it contains. Our method is divided into four main steps:

1. Map segmentation must first turn the input color map into a binary map in which characters are isolated from other features and characters.

2. In the component analysis step, connected components are generated and filtered.

3. Then, strings are created by grouping close connected components with similar properties.

4. Finally, detected strings are recognized by an OCR before being checked and written into a text file.

The chart flow of the full process is depicted in figure 2.

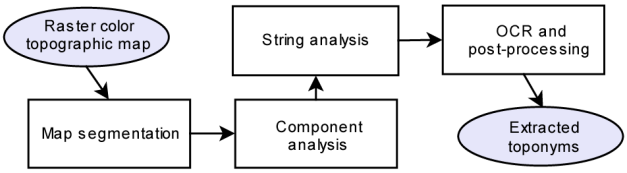

Figure 2. Algorithm chart flow.

3.2 Map segmentation

First, it is important to note that great care has to be taken regarding the quality of the scan process. In order to be well recognize during the whole process, the input image must be scanned with a resolution of at least 300dpi. This also insure that characters are large enough to be recognized by a classical OCR.

The first step of our technique consists in a binary classification of each pixel in order to eliminate pixel values that can definitively not be a part of a character. This assumption comes from the observation that characters on maps are printed using a very restricted and well defined color palette (eg. black for place names and blue for water network names).

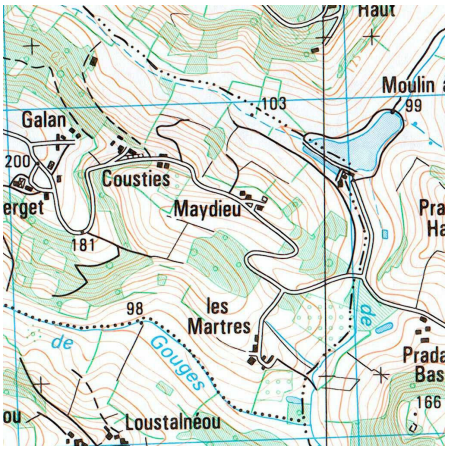

Figure 3. Sample of a color topographic map.

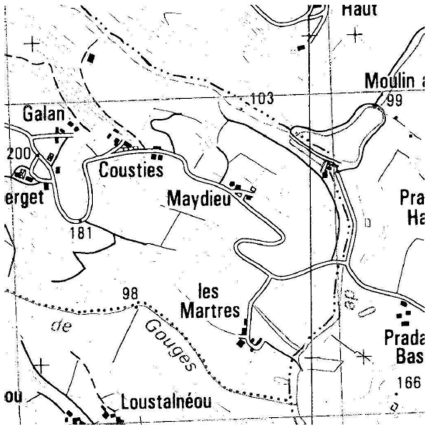

Figure 4. Binarization of the grayscale map with a threshold $\tau = 128$ followed by a morphological closing.

Our technique is based on a per-character processing, thus it requires that each character holds in a single connected component. There is a wide variety of topographic maps however maps are drawn with the concern of readability to be easily understood by humans. Thus, features and especially textual features are as much dissociated of other features as possible.

In numerous maps, a simple RGB-to-luminance (Y component) conversion followed by a binarization using a medium threshold is enough to obtain good results. However, for more complex maps, one could apply different kinds of preprocessing filters such as color segmentation using K-mean or morphological filters such as top-hat in the HSV space[1].

A morphological closing (erosion then dilatation) is applied on the obtained binary image in order to remove as many as possible of the small and thin linear features like contour lines or roads. Figure 4 shows the map 3 after the map segmentation processing.

3.3 Component analysis

Once we have a binary image with potential textual information, the next step is to label connected components. Connected components labeling in 8-connectivity is made using a classical 2-passes algorithm. Components are then indexed in a list. During this operation components bounding boxes are computed. Two pruning steps are then applied in order to remove as many outliers as possible:

- *Density pruning* - We call density of a connected component the percentage of black (object) pixels in the rectangle defined by the component bounding box. The first pruning step consists to remove components with density lower than a given percentage. We determined empirically that 25% was a good threshold for our maps.

- *Size pruning* - Let μ and ν be the mean height and height of the remaining components. We perform a pruning of each component c with $\max(height_c, width_c) < 0.5 \times \min(\mu, \nu)$ and $\max(height_c, width_c) > 2.0 \times \max(\mu, \nu)$. This filtering removes too large components while preserving some thin characters such as '`i`' or '`l`' regardless their orientation.

3.4 String analysis

We now define a set of non-oriented graphs describing words that are strings of characters where a node is a connected component. Graphs are constructed by linking pairs of components C_1 and C_2 with centroid (x_1, y_1) and (x_2, y_2) according to two criteria:

1. $|x_1 - x_2| < 1.5 \times \max\Big(width(C_1),\ width(C_2)\Big)$

2. $\dfrac{\max\Big(height(C_1), height(C_2)\Big)}{\min\Big(height(C_1),\ height(C_2)\Big)} < 2$

The first condition allows to connect neighbor components in the x-axis, while the second guarantees that the two components have similar heights.

Moreover, a second filtering needs to be applied on each non-oriented graph describing possible words. First, we have to compute the average height and width of each components. H_{mean} and W_{mean} represents the mean value of

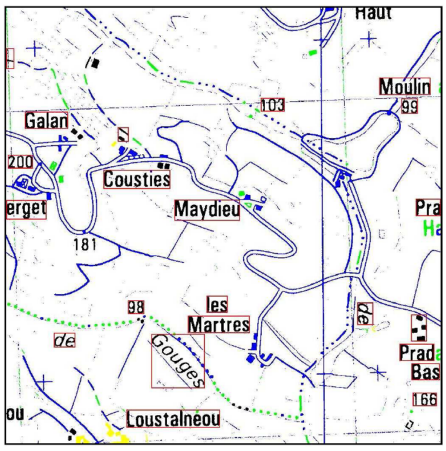

Figure 5. Strings of connected components remaining after the pruning pass are surrounded of a rectangle.

each characters. Finally, to validate each possible characters in a possible string (graph), the two following criteria have to be verified:

1. $H_G > \frac{H_{mean}}{2}$;
2. $W_G > (1.5 \times W_{mean})$.

where H_G is the height and W_G the width of the graph.

Figure 5 shows detected string components remaining on figure 4 after this step.

Once potential strings have been localized, the next part of the work is the recognition itself. In order to ensure that the text is oriented horizontally for being well recognized, we apply a skew detection using the average angle connecting the centroids of the components. Then, each string is rendered horizontally into a picture file which is used to feed an OCR software.

3.5 OCR and post-processing

The awaited OCR output is an ASCII text file containing the recognized words. In our implementation, we use GOCR by Joerg Schulenburg, an open source OCR program which can be used in command line.

Because there is no perfect character recognition, some characters will not be recognized at all (and they will be replaced by an underscore, see figure 6) or will be badly recognized. In order to filter spurious strings we process to last correction steps: i) strings must respect the following regular expression (expressed with the Unix format): `[a-zA-Z0-9_']{2,}` otherwise they are ignored; ii) we proceed some contextual corrections. For example, if the two first characters of the string are not digits, the string is not considered as un number (eg. elevation value) so '`0`' is replaced by '`O`' and '`5`' by '`S`'. Manual corrections may also be needed in order to correct characters that were not recognized by the OCR. Of course, a final step could consist, if available, in using a geographical dictionary to correct erroneous toponyms.

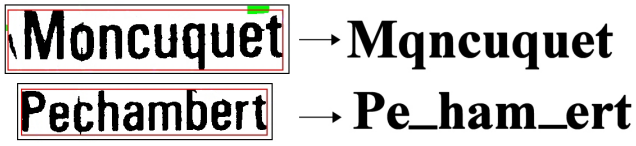

Figure 6. Examples of unrecognized characters using GOCR.

3.6 Merging strings

The final step of our technique consists to merge two neighbor strings which actually should be one unique toponym (see figure 7).

Figure 7. Examples of 2 strings forming one toponym.

The algorithm treats each string graphs by merging all labels that are closed each other, mostly parallel and with similar height.

Finally, the obtained toponym ASCII strings are easily exported to GIS formats like ESRI Shape files or Virtual Terrain Project (VTP) Location File format for further use.

4 Results and discussion

We implemented our recognition scheme as a plug-in for a GIS software specialized in features extraction from topographic maps. In order to evaluate the performance of our technique, experiments were conducted with a set of IGN topographic maps scanned at 500dpi. Table 1 sums up the results we obtained. We recall that *precision* is the percentage of correct detections in respect to the total, and *recall* rate is the percentage of correct detections in respect to the total number of detectable elements.

Considering the OCR recognition and the post-processing step, we obtain a recognition rate (ie. words perfectly recognized) near 89% using GOCR.

	#item	#detected items	#errors	Precision (%)	Recall (%)
Chars	1373	1271	48	96 %	92 %
Strings	266	242	42	91 %	85 %

Table 1. Toponym detection results.

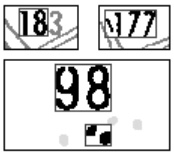

Figure 8. Most errors come from an over or under connection of characters.

However, as shown in figure 8 there are still missed strings due to over or under connected components. This kind of errors shows the limitations of this kind of technique based on connected components and the importance of the segmentation algorithm chosen in the first step of our method.

As each existing method of text extraction has specific conditions to be performed, the problem is to find the good criteria to compare our model with the others. The hypothesis based at the beginning of this project was that no assumption is made about the character font, size, color or orientation. As explained in section 2, previous approaches are quite specific and produce good results with specific conditions. In a sens, our technique suffer of the same kind of limitation, however it is well suited for many kind of scanned maps.

Figure 1 shows a 3D view of the extracted labels on a topographic maps (obtained without the usage of a dictionary).

5 Conclusion

An automatic method to extract and recognize the textual layer of scanned color topographic maps has been described. The proposed method is mainly based on image segmentation and connected component processing. At each step, pruning or corrections are done using empirical filtering. Detected potential text area are then processed by an OCR software from which returned ASCII strings are analyzed and eventually corrected while non conform strings are rejected. The obtained results relate quite good precision and recall rates. The strength of the method lie in its simplicity and efficiency. We think connected components methods are still very efficient for this kind of documents but in future work we will especially focus on a more robust segmentation algorithm in order to eliminate more linear features while preserving characters.

Acknowledgments

Maps are reproduced with the gracious authorization of the IGN - Institut Géographique National (France). Samples we taken from the map *TOP 25 #1940 O Série Bleue*.

References

[1] J. Angulo and J. Serra. Mathematical morphology in color spaces applied to the analysis of cartographic images. In *Proceedings of GEOPRO '03*, pages 59–66, 2003.

[2] R. Cao and C. Tan. Separation of overlapping text from graphics. In *Proceedings of ICDAR '01*, page 44, 2001.

[3] D. S. Doermann. An introduction to vectorization and segmentation. In *GREC '97*, pages 1–8, 1998.

[4] J. Duong, M. Côt, H. Emptoz, and C. Suen. Extraction of text areas in printed document images. In *Proceedings of DocEng '01*, pages 157–165, 2001.

[5] L. Fletcher and R. Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(6):910–918, 1988.

[6] J. Gllavata, R. Ewerth, and B. Freisleben. A robust algorithm for text detection in images. In *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, Rome, Italy*, pages 611–616, 2003.

[7] J. Gllavata, R. Ewerth, and B. Freisleben. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In *Proceedings of ICPR '04*, pages 425–428, 2004.

[8] A. Khotanzad and E. Zink. Contour line and geographic feature extraction from USGS color topographical paper maps. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(1):18–31, 2003.

[9] S. Levachkine. Raster to vector conversion of color cartographic maps. In *GREC*, pages 50–62, 2003.

[10] S. Levachkine, A. Velálzquez, V. Alexandrov, and M. Kharinov. Semantic analysis and recognition of raster-scanned color cartographic images. In *GREC '01: Fourth International Workshop on Graphics Recognition Algorithms and Applications*, pages 178–189. Springer-Verlag, 2002.

[11] L. Li, G. Nagy, A. Samal, S. Seth, and Y. Xu. Cooperative text and line-art extraction from a topographic map. In *Proceedings of ICDAR '99*, pages 467–470, 1999.

[12] C.-W. Liang and P.-Y. Chen. DWT based text localization. In *International Journal of Applied Science and Engineering*, 2004.

[13] M. Pierrot-Deseilligny, H. L. Men, and G. Stamon. Characters string recognition on maps, a method for high level reconstruction. In *Proceedings of ICDAR '95*, page 249, 1995.

[14] H. Tran, A. Lux, H. L. N. T, and A. Boucher. A novel approach for text detection in images using structural features. In *Proceedings of ICAPR '05*, pages 627–635, 2005.

[15] A. Velázquez and S. Levachkine. Text/graphics separation and recognition in raster-scanned color cartographic maps. In *Proceedings of GREC '03*, pages 63–74, 2003.