MVA2011 IAPR Conference on Machine Vision Applications, June 13-15, 2011, Nara, JAPAN

Automatic Road Extraction from Printed Maps

Sebastien Callier and Hideo Saito
Keio University
3-14-1 Hiyoshi, Kohoku-ku
Yokohama, Japan
{scallier,saito}@hvrl.ics.keio.ac.jp

Abstract

Raster maps are widely available in the everyday life, and can contain a huge amount of information of any kind using labels, pictograms, or color code e.g. But due to those overlapping features, it's not an easy task to extract roads from those maps. Many methods use user input to achieve this goal.

In this paper we focus on an automated method to extract roads by using color segmentation and linear features detection to search for seed points having a high probability to belong to roads. Those seeds are then expanded before choosing to keep or to discard the extracted element.

Keywords: Road Extraction, Raster Map, Seed Points

I. INTRODUCTION

Raster maps are one of the most used tools to know our position, and the direction to choose. However, the main drawback of those maps is their lack of interactivity. We can find this interactivity with services of Global Positioning System that are now really common but they are not as complete as specialized maps. The idea is to extract road network and intersection data from maps such as tourism maps and to be able to match them with other maps where GPS coordinates are known – because roads are the most common and often the main features among the different maps to locate ourselves.

Fig. 1. Example of raster map with a lot of overlapping features.

Some difficulties to face in road extraction are:

- Raster maps usually contain a lot of information disturbing the road recognition such as street names, icons, topologic lines etc. like in Fig.1.

- The quality of the different pictures is really variable depending on their origin – photography or scanned map – and also on the quality of the tool used.

To extract features from raster maps, some research relies on color segmentation whether in RGB, HSV or in grayscale space. Khotanzad et al. in [6] utilize color segmentation with user input to extract contour lines in USGS topographic maps. Henderson et al. in [7] have a method using preset color index to produce a set of rules for classification of the pixels. Or later, Chiang et al. in [1] are also using a color segmentation followed by K-mean algorithm to reduce the number of different colors to 15. Then, there is a manual step where the user has to select the road color layers by clicking on them. Another example of semi automated solution based on color could be Cao et al. [2] who use a preset grayscale threshold in order to detect the background of raster maps.

The first drawback of color segmentation-based method appears if the road color is also the background color, which appends in some maps and also in hand written maps. Furthermore, choosing an arbitrary number of colors isn't reliable enough to assume this will work with a random map.

Some other works utilize different separation like Luo et al. [8] who try to separate roads and text labels in simple maps. Or Hai et al. in [9] who separate pixels into three possible groups: road, non-road and noise regions. Noise region regions includes texts, symbols etc. that overlap with both road and non-road region. Chen et al. in [12] interprets images of Chinese land register maps using character recognition to extract text, and line detection.

Chiang et al. [3] aims to learning different kind of maps – google maps, yahoo maps, live maps for example. From a set of maps, they learn to classify maps using luminance-boundary histograms. Except from the learning part everything is then automated. But tourism maps are very different one from another and use unique features and we want a road extraction method working on those particular maps. This process isn't suitable in that case.

In this paper, we will present a fully automated method to extract roads from raster map without any previous knowledge about the colors or user input. One of the final goals is to be able to analyze a photographed map with a mobile phone in order to locate it and use this map with GPS data. Our method will first reduce the overall number of color in the raster map by using a mean shift algorithm. Then, a search of linear features in every direction around each pixel will be used to have a first probability for every pixel to belong to the background or to the road. The pixels with the highest probability to belong to road are used as seed points to extract them. Finally, by processing on color histograms, we will identify main road colors and background colors to retrieve some undetected pieces of road.

II. Method

The input data is an image of a printed map taken by a camera or a scanner. From the input image, we only want to keep the road pixels. Our method explained in Fig.2. consists first in reducing the noise in the raster map. Then, detecting the linear features to create a probability map for every pixel to belong to a road or not. The selected pixels are the seed points to build the road network. Finally, with a color histogram we extract some more pixels to refine the results. Every part of the method will be explained in more details in the following section.

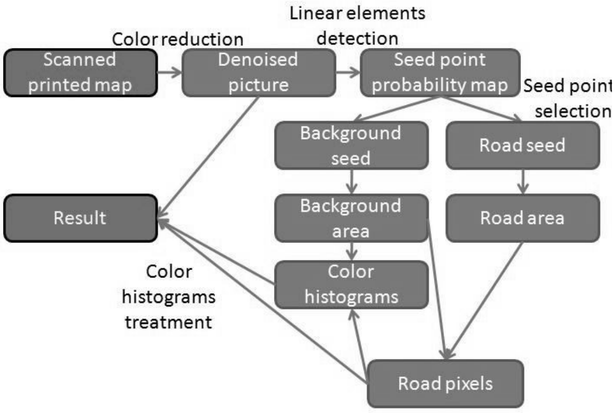


Fig.2. Overview of the proposed method

A. *Color reduction*

As we have already mentioned, due to the many different sources of raster maps, we can't have a precise knowledge of the quality of the picture we will have to analyze. To try and reduce the noise in the picture colors, the first idea would be to use a Gaussian or median filter. However, such filters imply a substantial loss of information about the position of the roads and their edges.

That is why our choice heads to mean shift method [10]. That reduces the overall number of colors without changing drastically the data. We set it with a 3 pixel radius; i.e. all pixels within this radius around the considered pixel that have a distance below 20 in the BGR color space are selected and get the same color as the this pixel.

B. *Linear elements detection*

To find the roads we will try to find which pixels have the greatest probability to belong to the roads, and those points will be used as seeds.

In order to achieve this detection, we will consider the 30 different possible directions of line in a 15x15 region of interest around every pixel – Fig.3. gives some examples of those lines. If we find a line where all the pixels have almost the same color, then we will consider it as a possible linear feature in that direction.

We create a table of the same size as the picture to memorize the number of direction in which we have detected linear features around each point. Every time we find a line of almost uniform color we increment the corresponding value in the table. A graphical representation of the result is shown in Fig.4(b). We already can see some pieces of road lines and uniform background areas appearing. We have now to classify those points before performing the seeded region growing.

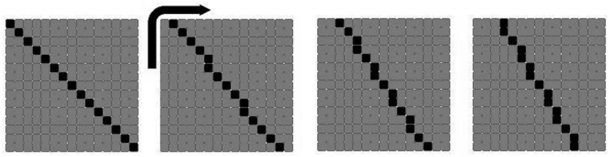
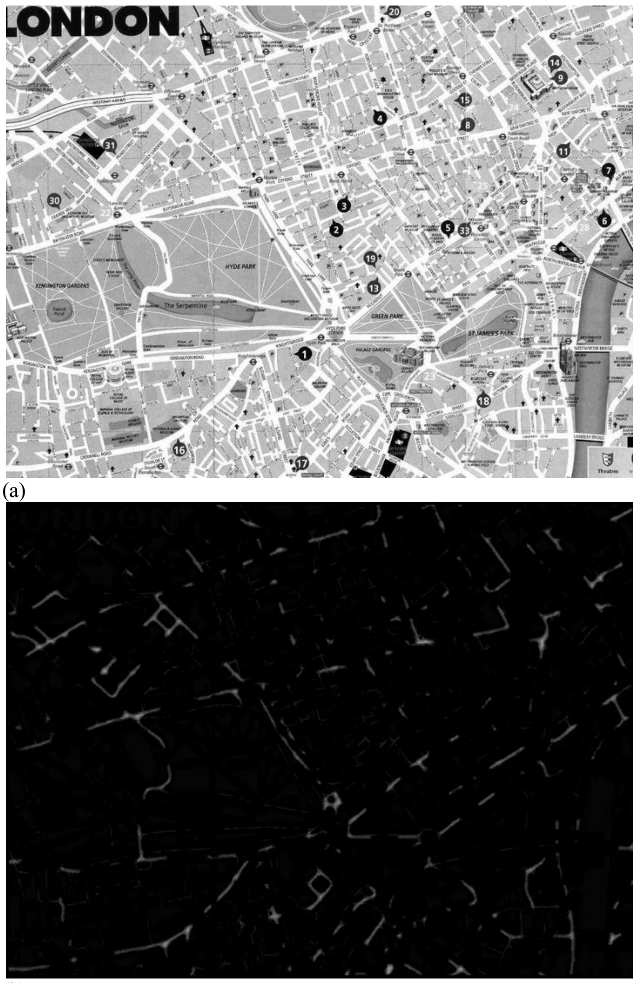


Fig.3. 4 Linear filters out of the 30

C. *Seed points selection*

In the table we created, the maximum values are most likely corresponding to pixels belonging to the background of the picture. Indeed, that means all the pixels in a 15x15 area are almost of the same color. We apply a seeded region growing process to the pixels corresponding to high values using color similarity and spatial connectivity. All the areas we retrieve after this treatment are considered as background points and discarded as road possible seed points.



(a)

(b)

Fig. 4. (a) Input map
(b) Output of the linear filters

Low values might be corners, as well as text elements, pictograms or even roads whose detection is disturbed by high noise. There are no particular treatments for those points while they are not conclusive.

The medium values have a great chance to belong to a road, or at least a linear element, and so they are suitable seed points. We apply a seeded region growing and as soon as the expanded area is in contact with a discarded area, we erase it.

At this point, we already retrieve an interesting amount of pieces of road network separated one from the other because of the noise. But some parts of road aren't extracted for two main reasons:

- Too small pieces of road – between the different words of the road name e.g. – or turning roads are more difficult to locate with this technique.
- Too big roads with huge crossing that could be confused with background areas are automatically discarded.

To add those part of road network that we can't neglect – especially the main roads with important crossings, we will now use the color of the selected pixel on one hand and of the discarded on the other hand.

D. *Color histogram treatment*

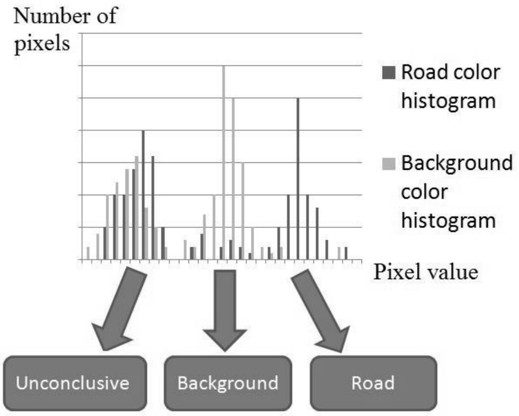


Fig.5. Color histograms' treatment principle

Two color histograms are created in order to retrieve some undetected road pixels. The first one contains road pixel colors and the other discarded pixels. Main peaks in road pixels are selected as road color if there isn't a peak around the same position in discarded pixels. We do the same with discarded points, to get rid of miss detected points. This principle is summarized in Fig.5. which is a simplified representation of the real three dimensions histogram.

III. RESULTS

We used our method and the one presented in [1] – which consist in using K shift, followed by the K mean method to get 15 different color layers before selecting the road layers – to compare the results on low quality scanned maps. We will focus on two values to compare the results:

- The correctness: i.e. the percentage of road pixels well detected.
- The completeness: i.e. the percentage of detected pixel that really belong to roads.

We did those comparisons on a few low quality maps by selecting the filters to get the highest completeness first, then the highest correctness for the method described in [1].

The average completeness is around 80~85% and the correctness is usually above 99% for our method with the different tourism maps we used. The method presented in [1] only enables to retrieve around 60% of the road pixels with 50% correctness.

Fig. 6. Roads automatically extracted from Fig.4.a.

Around 80% of the non detected road pixels in our method are a result of text labels – manly road names – the last 20% are a result of overlapping features or too high noise even after the K mean. Some results are shown in Fig.6. and Fig.7.

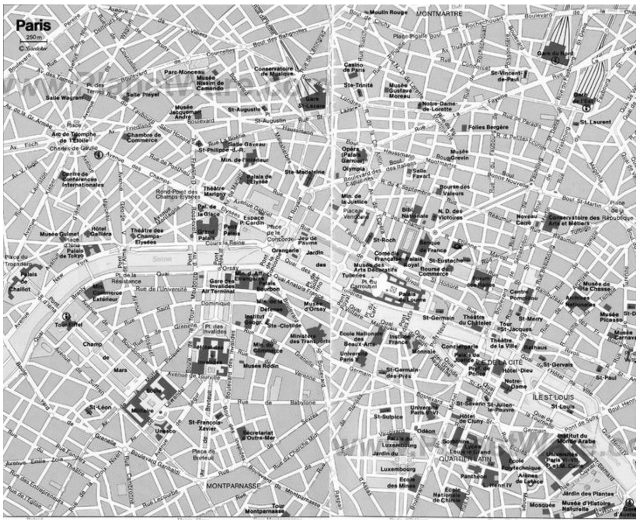
IV. CONCLUSION AND FUTURE WORK

In those examples, almost all the pixels of the road color are extracted. Even if the holes in the roads due to the different labels remain, this kind of result may be enough to extract road intersections and to try to match up this map with a GIS database of crossroads.

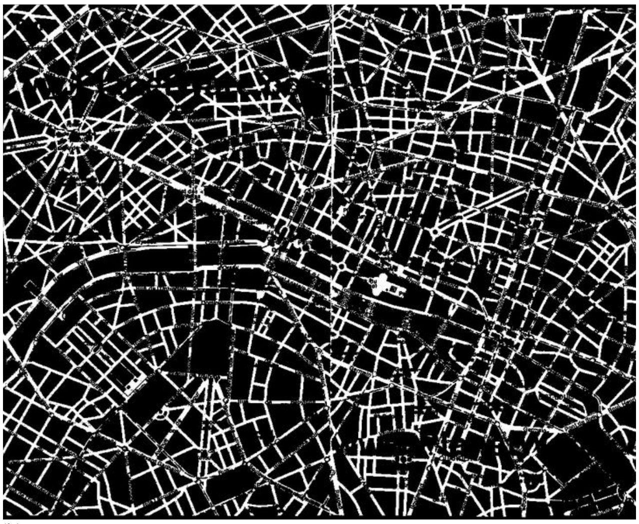
We can obtain this kind of results with almost all the tourism map we used as test map. The only maps causing troubles was a map using the road color in many little pictograms elements and drawing which was decreasing the correctness to around 85%.

In this paper we don't take into account the final shape of the extracted roads. Dealing with those shapes could be a way to get rid of some wrong pixels while filling to holes in the roads due to the overlapping elements.

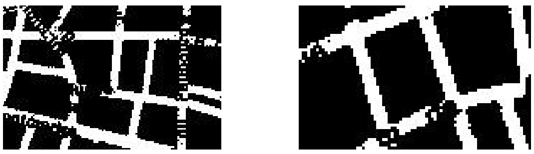
Our next important step before trying to match the maps will be to try to identify the intersections and retrieve their accurate position in the map. From this and then using a key point matching algorithm like described in [5], from a database on the intersection coordinates we could identify the position of the raster map.



(a)



(b)



(c)

Fig.7. (a) Raw scanned map of Paris
(b) Detected roads
(c)Close up view of extracted roads

ACKNOWLEDGEMENTS

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for high-Level Global Cooperation for Leading Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology and Grant-in-Aid for JSPS Fellows in Japan.

REFERENCES

[1] Chiang Y-Y, Knoblock C a. A Method for Automatically Extracting Road Layers from Raster Maps. *2009 10th International Conference on Document Analysis and Recognition*. 2009:838-842.

[2] R. Cao and C. L. Tan. Text/graphics separation in maps. In Proceedings of the Fourth International Workshop on Graphics Recognition Algorithms and Applications, pages 167–177, 2002.

[3] Chiang Y-Y, Knoblock C a. Classification of raster maps for automatic feature extraction. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*. 2009:138.

[4] Marikhu R, Dailey MN, Makhanov S, Honda K. A Family of Quadratic Snakes for Road Extraction. *Computer Graphics and Image Processing*.

[5] Uchiyama H, Saito H, Servières M, Moreau G. Camera tracking by online learning of keypoint arrangements using LLAH in augmented reality applications. *Virtual Reality*. 2010.

[6] A. Khotanzad and E. Zink. Contour line and geographic feature extraction from USGS color topographical paper maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence.25(1):18-31, 2003*

[7] Henderson TC, Linton T. Raster Map Image Analisys. *2009 10th International Conference on Document Analisys and Recognition.* 2009:376-380

[8] Huizhu Luo and Rangachar Kasturi. Improved Directional Morphological Operations for Separation of Characters from Maps/Graphics. In K. Tombre and A. K. Chhabra, editors, *Graphics Recognition – Algorithms and Systems,* volume 1389 of *Lecture Notes in Computer Science*, pages 35-47, Springer-Verlag, April 1998

[9] Hai T, Bao Y-lu. Road Extraction from Color Raster Urban Triffic Map, *2008 International Conference on MultiMedia and Information Technoligy,* 2008;1(1):381-384

[10] Comaniciu D. and Meer P. Mean Shift: a Robust Approach Toward Feature Space Analysis. *IEEE Transactions on PAMI*, 24(5):603-619, 2002

[11] Leyk S. and Boesch R. Colors of the Past: Color Image Segmentation in Historical Topographic Maps Based on Homogeneity, *GeoInformatica,* 14(1):1-21, 2010

[12] Chen L, Liao H, Wang J, Fan K, 1999 Automatic Data Capture for Geographic Information Systems. *IEEE Trans Syst Man Cybern* C 5(2):205-215