A Method for Automatically Extracting Road Layers from Raster Maps

Yao-Yi Chiang and Craig A. Knoblock
University of Southern California
Department of Computer Science and Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292
yaoyichi, knoblock@isi.edu

Abstract

To exploit the road network in raster maps, the first step is to extract the pixels that constitute the roads and then vectorize the road pixels. Identifying colors that represent roads in raster maps for extracting road pixels is difficult since raster maps often contain numerous colors due to the noise introduced during the processes of image compression and scanning. In this paper, we present an approach that minimizes the required user input for identifying the road colors representing the road network in a raster map. We can then use the identified road colors to extract road pixels from the map. Our approach can be used on scanned and compressed maps that are otherwise difficult to process automatically and tedious to process manually. We tested our approach with 100 maps from a variety of sources, which include 90 scanned maps with various compression levels and 10 computer generated maps. We successfully identified the road colors and extracted the road pixels from all test maps with fewer than four user labels per map on average.

1. Introduction

Numerous maps are available in raster format and are easily accessible from the Internet. For example, the scanned USGS topographic maps can be downloaded from the Microsoft Terraserver and many other information rich raster maps can be found in map repositories such as the University of Texas Map Library. [1] To utilize the information in the raster maps, in our previous work, we developed a technology to first identify the road intersection templates in the raster maps [4] and then match the set of road intersection templates with another set of road intersection templates from a georeferenced data set (e.g., vector data) [2] to identify the geocoordinates of the maps and align the maps with the georeferenced data. For the automatic road intersection extraction process, in [4], we employed a histogram analysis approach to extract the foreground pixels from the raster maps and utilized a text/graphics separation algorithm [1] to extract the road lines. However, due to the complexity of maps and noise introduced in producing the maps in raster format (e.g., scanning), it is difficult to separate the foreground pixels from raster maps automatically; even manual work requires tedious work to select the colors that represent the foreground pixels in a scanned map. Without properly extracting the foreground pixels, the road lines cannot be extracted since the text/graphics separation algorithm will not have a correct input to work with.

In this paper, we present a supervised technique that requires minimum user input for extracting the road pixels from raster maps. Our technique automatically identifies the colors that represent roads in the raster map by analyzing user labels (i.e., rectangular image areas labeled by the user). The only requirement of each user label is that the label needs to be approximately centered at a road intersection or a road line segment. The user label does not have to contain only road pixels, which makes the user-labeling task easier and practical. The identified road colors are then used to generate a color filter to extract the road pixels from the raster map. The extracted road pixels can be refined using simple morphological operations or sent to a text/graphics separation component to remove the characters if the road colors are also used to draw other map features, such as text. The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 describes our approach to extract the road pixels from a raster map. Section 4 reports on our experimental results, and Section 5 provides the discussion and future work.

2. Related Work

Much research work has been performed in the field of extracting graphic features from raster maps, such as extracting road lines and contour lines. To extract the graphic features, a binarization step is commonly used to first extract the foreground pixels, and there are a variety of algorithms that are then used to extract features from the foreground pixels. Our approach in this paper is a binarization step that can be used for road extraction if the road colors

[1] http://www.lib.utexas.edu/maps/

are only used for roads, which is generally more efficient than analyzing the geometry of the extracted foreground pixels to extract roads [1, 9]. If the road colors are used on other map features, our method can be used to replace the binarization steps in existing text/graphics techniques to reduce the number of foreground pixels that need to be processed, which makes it possible to handle more complex maps. In this section, we review different feature extraction techniques and focus on their binarization steps.

Cao et al. [1] utilize a preset grayscale threshold to remove the background pixels from raster maps and then detect text labels from the remaining foreground layers of the maps. Li et al. [9] utilize a fixed color filter to extract the "black layer" from the USGS topographic maps and then work on the black layers to extract and rebuild the text labels and lines. Pouderoux et al. [10] also use a preset grayscale threshold to generate a binary map for their toponym recognition algorithm, and the authors also note that additional methods such as the K-means color segmentation should be adopted to process more complex maps. Habib et al. [7] work on the raster maps which contain only road lines to extract road intersections, and hence their binarization step is simply an edge detector. These approaches each handle a specific type of map that can be processed by their default binarization methods. Our extracted road pixels can be used as the input to these feature extraction techniques that employ simpler binarization steps to further extract specific graphic features (e.g., road vectors or road intersections) or to refine the road lines if needed.

In our previous work [4], we utilize a histogram analysis method to automatically separate the foreground pixels from the raster maps and then identify the road pixels and reconnect the road lines. Although the histogram analysis method does enable us to generate different binarization thresholds for different maps, the histogram analysis does not handle scanned maps well since the noise introduced in the scanning process is sometimes difficult to remove automatically. Our method in this paper includes user training and is capable of handling more diverse types of maps, especially scanned maps.

Other techniques include user training in their binarization steps. Salvatore and Guitton [11] use color classification methods as their first step to extract contour lines from topographic maps. Khotanzad et al. [8] utilize a color segmentation method with user annotations to extract the contour lines from the USGS topographic maps. Chen et al. [3] extended the color segmentation method in [8] to handle common topographic maps (i.e., not limited to the USGS topographic maps) using local segmentation techniques.

These techniques with user training are generally able to handle more complex maps. However, their user training processes are complicated and laborious. In [11], their technique requires manual examinations on the input maps to generate a proper set of color thresholds for a specific set of maps. In practice, a scanned topographic map can have thousands of colors and the thresholds for different topographic maps may vary depending on the quality of the map, which makes the manual examination repetitive work. In [3, 8], the users are required to labeled all combinations of line and background colors. For comparison, our approach in this paper requires that the user label only a few road areas, which is simpler and more straightforward.

3. Automatic Extraction of Road Pixels from Raster Maps

The input of our technique for extracting the road pixels is either a scanned map or a digital map (maps generated directly from vector data), and the output is the road pixels of the raster map. We first apply color segmentation algorithms to quantize the input image. The color segmentation algorithms reduce the number of colors that we need to verify for identifying the road colors in the next step. Then, the user labels several road areas in the quantized map. From the user labels, we identify a set of road colors and generate a color filter to extract the road pixels from the raster map. The details of each step and the labeling criteria are described in the following subsections in turn.

3.1. Color Segmentation

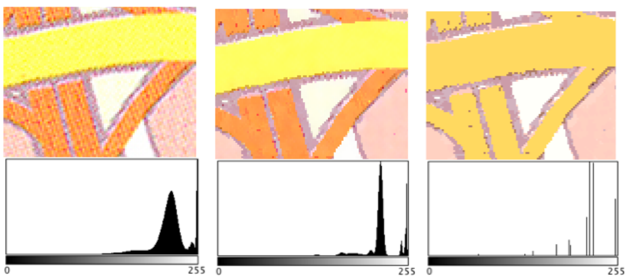
Raster maps usually contain numerous colors. For example, the sample raster map shown in Figure 1(a) has 285,735 colors, which is difficult for identifying road colors in the latter steps. The two color segmentation techniques we use to reduce the number of colors in the input image are the Mean-shift [5] and the K-means [6] algorithms.

The Mean-shift algorithm merges two colors into one by considering their distance in the color space (we use a color distance of 25 in the red, blue, and green color space in our method) as well as in the image space (we use a spatial distance of 3 pixels in our method), which helps to preserve object edges while performing the color segmentation. Figure 1(c) shows a portion of the sample map where the number of colors is reduced by 50% (i.e., 155,299 colors) by applying the Mean-shift algorithm. Next, we apply the K-means algorithm to ensure that the final quantized map has a number of colors that is not larger than K (we use K=10 in our method). Figure 1(d) shows a portion of the sample map where there are only 10 different colors remaining after applying the K-means algorithm.

3.2. User Labeling

The number of colors in the quantized map is less than or equal to 10 after the color segmentation (i.e., K=10), so the number of road colors is lower than 10. In this user-labeling step, the user needs to provide a user label for each road color in the quantized map. A user label is a rectangle

(a) An example scanned map



(b) The original map (c) Use the Mean-shift algorithm only (d) Use both the Mean-shift and K-means algorithms

Figure 1. An example map and the quantized maps with their grayscale histograms

that should be large enough to cover a road intersection or a road segment. To label the road colors, the user first selects the size of the label. Then, the user clicks on the approximate center of a road line or a road intersection to indicate the center of the label. The user label should be (approximately) centered at a road intersection or at the center of a road line, which is the constraint we exploit to identify the road colors in the next step. For example, the right side of Figure 2 shows a typical labeling result to extract the road pixels from the quantized map shown in the left. The two user labels cover one road intersection and one road segment, which contain the two road colors (i.e., yellow and white) in the quantized map.

3.3. Road Colors Identification

To identify the road colors, we first decompose each user label into color images so that every color image each contains only one color from the user label. For example, the user label shown in the top-right of Figure 2 is decomposed into six different color images shown in the top of Figure 3. For each color image, we extract the skeletons of every connected object in the image using the thinning operator, and apply the Hough transformation [6] to identify a set of Hough lines from the connected objects' skeletons. Since the image center of a user label is the center of a road line or a road intersection, we compute the average distance

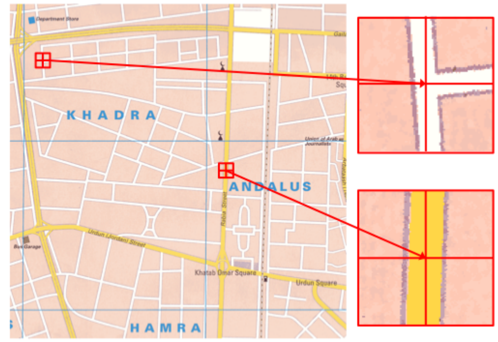


Figure 2. The quantized map and user labels

between the detected Hough lines to the image center for each color image as a measure to determine if the color pixels (i.e., non-black pixels) in an image represent roads in the raster map.

Figure 3 shows the detected Hough lines of each image, where the Hough lines that are within a distance threshold to the image centers are drawn in red and others are drawn in blue (this distance threshold is only used in the paper to help explain the idea). We can see that the images that contain road pixels (image 3, 4, and 5) have more red lines than blue lines and hence the average distances between their Hough lines to their image centers are smaller than the other images. Therefore, *the image that has the smallest average distance is first classified as the road-pixel image (i.e., the color pixels in the image are road pixels in the raster map).* The other color images that have their average distances within one pixel to the smallest average distance are also classified as road-pixel images. This criterion allow the user label to be a few pixels off (depending on the size of the user label) from the actual center of the road line or road intersection in the map, which makes the user labeling easier. In our example, image 5 has the smallest average distance, so image 5 is first classified as a road-pixel image. Then, since image 4 is the only image that has its average distance within a one-pixel distance to the smallest average distance, image 4 is also classified as a road-pixel image.

The Hough transformation method relies on the numbers of detected Hough lines so if a color that is used on roads has a smaller number of pixels compared to the major road colors, such as the image 3 shown in Figure 3, the image will not be classified as a road-pixel image using the Hough transformation method. Therefore, we employ a template matching algorithm to check if any of the color images that are not classified as road-pixel images using the Hough transformation method (i.e., image 0 to 3) is a road-pixel image. We first generate a road template image using the already classified road-pixel images. In our example, the road template image is the combination of image 4 and 5 as shown in Figure 4. In the road template image, the color pixels are road pixels and the black pixels are the

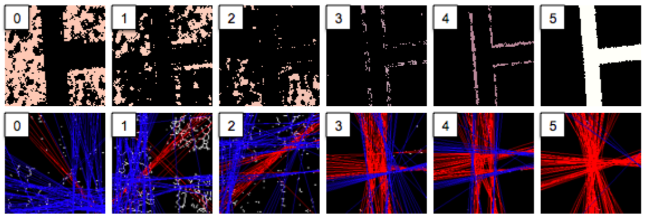


Figure 3. The color images and identified Hough lines (background is shown in black)

Figure 4. The road template image

background. Next, the road template image is used to evaluate image 0 to 3 iteratively. For a color pixel, C(x, y), in a given color image to be evaluated, we search a 3-by-3 pixel neighborhood centered (x, y) in the road template image to detect if there exists any road pixels (i.e., color pixels). If one or more road pixels are detected, the pixel C(x, y) is marked as a road pixel since it is spatially near one or more road pixels. After every color pixel in a given color image is examined, if more than 50% of the color pixels in that image are marked as road pixels, the given image is classified as a road-pixel image.

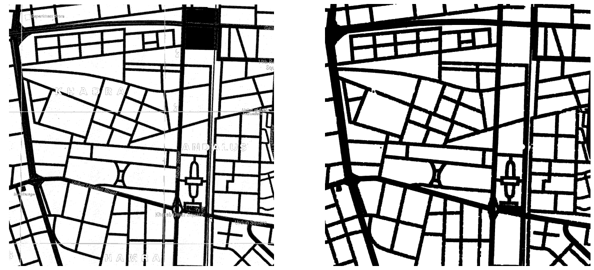
Every user label is processed using the method described in this section and a set of road-pixel image is identified for each label. The colors in the road-pixel images (i.e., the identified road colors) are then used in the next step to generate a color filter.

3.4. Road Pixel Extraction

We generate a color filter using all identified road colors and then scan the quantized map to extract the road pixels. The colors in the quantized map that are not in the color filter are filled with white pixels and the others are filled up with black pixels. Figure 5(a) shows the extracted road pixels of Figure 1(a). The extracted road pixels then can be refined using morphological operations (e.g., the erosion and dilation operators) to remove solid areas and reconnect lines as shown in Figure 5(b).

4. Experimental Setup and Results

We tested our approach on seven sets of maps from various sources. The first three sets of maps were 30 topographic maps (600x600 pixels each) from USGS covering three different U.S. cities. The USGS topographic maps are commonly used in the experiments of map feature extraction research [9, 8], which contain noise introduced during the map production processes, and different areas of the topographic maps have very different degrees of noise and different color usages. In addition to the USGS maps, we extended our experiments to test our technique on processing commonly accessible scanned maps using three sets of 60 maps (2000x2000 pixels each) cropped from three different scanned maps (350dpi) covering Bagdad, Iraq. The three scanned maps were published from different publishers[2] and different legends were used in the maps. Moreover, since the original paper maps have been folded, there were folding lines on the paper maps that caused inevitable shadows and color differences on different areas of the scanned raster maps. Therefore, even the maps cropped from the same scanned map have various color usages. The last set of maps were 10 digital maps (1500x1500 pixels each) from Rand McNally covering St. Louis, U.S., which were tested in our experiment because the digital maps have white background and a majority of roads on the maps have light colors. As a result, the roads of light colors are usually misclassified as background pixels using our automatic approach, which analyzes the grayscale histogram [4].



(a) The extracted road pixels using the identified road colors

(b) The extracted road pixels with post processing

Figure 5. Road extraction results

In the experiment, the user utilized our implemented system to label road areas in each test map for extracting the road pixels from the map. Our implemented system generated a road extraction result after every user labeling, and the user can preview the result in real-time to decide if more labeling needs to be done for a better extraction result. A successful extraction is where the user can use our implemented system to extract the majority of the road colors in a raster map without examining the image histogram or selecting individual colors. The majority of the road colors means that if not every road color in the raster map is identified, the road topology can still be rebuilt using morphological operations (e.g., the erosion and dilation operators). Figure 5 shows a typical result with the majority of road colors were identified using our system and the result after morphological operations to remove solid areas and small connected objects.

We successfully extracted the road pixels from all test maps, and the average number of user labels per map for

[2]The paper maps are produced by Gizi Maps, Gecko Maps, and International Travel Maps

Map Set	Map Count	Avg. User Labels	Avg. Computation Time (s.)
1. USGS topo. maps	19	1.5	1.5
2. USGS topo. maps	5	1.6	1.2
3. USGS topo. maps	6	1.8	1.8
4. Scanned maps	12	2.7	2.7
5. Scanned maps	18	2.1	3.3
6. Scanned maps	30	2.6	4.2
7. Rand McNally maps	10	3.9	2.4

Table 1. Experimental results of average user labels and computation time per map

every test set was under four as shown in Table 1. After the automatic extraction process, the user can then examine the results and decide if any post-processing needs to be applied. The extraction results of the scanned Iraq maps need only morphological operations for post-processing; the operations can be found in the implementation of [1]. The extraction results of the Rand McNally and USGS topographic maps require text/graphics separation techniques to further remove the characters, since their road colors are also used for characters. For the USGS topographic maps, many of the extraction results include contour lines. This is because the brown pixels in the USGS topographic maps are used on the contour lines and roads. The contour lines can be removed by the parallel-pattern tracing method in [4].

The system was built using Microsoft Visual Studio 2005 on a Windows 2003 Server with Intel Xeon 1.8 GHz Dual Processors and 1 GB of memory. Tabel 1 shows the average computation time per map in each set. The computation time was the computer time needed to process each user label and produce the extraction result for an input map, which mainly depended on the size of the map and the number of labels. The average computation time for a map was below five seconds, which allows us to generate a real-time preview of the result after each user labeling.

5. Discussion

In this paper, we present an approach to automatically extract road pixels from raster maps. Our approach exploits the constraint that the user labels are centered at road intersections or road lines to identify road colors for extracting the road pixels from the raster maps automatically. The experiments show successful road-pixel extractions from all of the 100 test maps with less than four user labels per map in average. The extraction results enable feature extraction techniques such as the road intersection extraction or road vectorization to handle more diverse types of maps. In the future, we plan to utilize a content-based image retrieval approach to automatically reuse the identified color filters for road extraction task on similar maps.

6. Acknowledgement

This research is based upon work supported in part by the University of Southern California under the Viterbi School Doctoral Fellowship, in part by the United States Air Force under contract number FA9550-08-C-0010.

The U.S. Government is authorized to reproduce and distribute reports for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of any of the above organizations or any person connected with them.

References

[1] 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. Springer-Verlag, 2002.

[2] C.-C. Chen, C. A. Knoblock, and C. Shahabi. Automatically and accurately conflating raster maps with orthoimagery. *GeoInformatica*, 12(3):377–410, 2008.

[3] Y. Chen, R. Wang, and J. Qian. Extracting contour lines from common-conditioned topographic maps. *IEEE Transactions on Geoscience and Remote Sensing*, 44(4):1048–1057, 2006.

[4] Y.-Y. Chiang, C. A. Knoblock, C. Shahabi, and C.-C. Chen. Automatic and accurate extraction of road intersections from raster maps. *GeoInformatica*, 13(2):121–157, 2008.

[5] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[6] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[7] A. Habib, R. Uebbing, and A. Asmamaw. Automatic extraction of road intersections from raster maps. Technical report, Center for Mapping, 1999.

[8] 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.

[9] L. Li, G. Nagy, A. Samal, S. C. Seth, and Y. Xu. Integrated text and line-art extraction from a topographic map. *International Journal of Document Analysis and Recognition*, 2(4):177–185, 2000.

[10] J. Pouderoux, J. C. Gonzato, A. Pereira, and P. Guitton. Toponym recognition in scanned color topographic maps. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, volume 1, pages 531–535, Sept. 2007.

[11] S. Salvatore and P. Guitton. Contour line recognition from scanned topographic maps. In *Proceedings of the Winter School of Computer Graphics*, 2004.