SRC: A Fully Automatic Geographic Feature Recognition System

Weiwei Duan
Computer Science Department
University of Southern California
weiweidu@usc.edu

Yao-Yi Chiang
Spatial Sciences Institute
University of Southern California
yaoyic@usc.edu

1 INTRODUCTION

Historical maps store abundant and valuable information about the evolution of natural features and human activities, such as changes in hydrography, the development of the railroad networks, and the expansion of human settlements. Such knowledge represents a unique resource that can be extremely useful for researchers in the social and natural sciences to better understand how human and environment have evolved over time. Fortunately, a large amount of historical maps have been scanned in high resolution by many organizations. For example, the United States Geological Survey (USGS) has scanned and released more than 200,000 historical maps in the TIFF format.

There are many researchers working in geographic feature from scanned maps to make the map content accessible in an analytic environment, such as the road recognition system [1], the elevation contour lines recognition system [3], and the text recognition system [2]. However, these existing systems require human interaction or prior knowledge of the input map. For example, [1] requires users to label the samples of road lines and road intersections. [3] is a system for reconstructing contour lines from scanned maps, in which users are required to choose the size of an image processing mask in order to reconnect broken contour lines. Besides user labeling, some systems use prior knowledge to separate map layers. For example, Henderson and Linton [4] exploit the color index of USGS maps and use it to separate layers. The need of user input for sampling or labeling and prior knowledge is the persistent limitation to process large numbers of maps in current recognition systems.

In this paper, we propose a prototype of a fully automatic geographic feature recognition system, which can process multi-edition, multi-scale historical maps in archives using semantic knowledge. Our system exploits the fact that map content does not change significantly between map editions to automatically generate accurate training samples (e.g., sample images of railroads and buildings) for building an accurate recognition model for each map. The following sections explain the challenges in building a fully automatic geographic feature recognition system, the architecture of our system, the preliminary results.

2 CHALLENGES IN BUILDING A FULLY AUTOMATIC RECOGNITION SYSTEM FOR MAPS

There are two main challenges in building a fully automatic geographic feature recognition system for scanned maps. The first challenge is that human interaction and prior knowledge makes processing large numbers of maps impractical. To solve this challenge, our system uses existing vector data to label the locations of the geographic feature in maps automatically instead of user labeling. Figure 1 is an example of the vector data locates the location of railroads in the map.

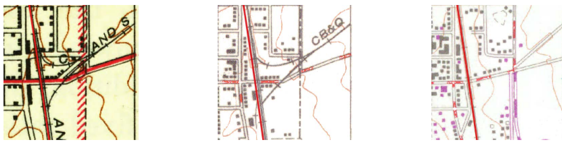
The second challenge is that map archives contain multi-edition and multi-scale maps. One of the difficulties of processing multi-edition maps is that the cartographic symbols for the same geographic feature may be different in different map editions. For the machine learning techniques used in traditional image recognition systems, e.g., the Support Vector Machine (SVM), the users have to define different feature descriptors to represent different symbols. To process different editions without defining different feature descriptors, our system employs the Convolutional Neural Networks (CNN) and generates new training samples automatically for each of the input maps. The inputs of the CNN in the training process are the labeled images. During the training, the CNN learns the feature descriptors to represent images by itself. CNN models have showed impressive performance in image recognition [5–7].

Another difficulty in processing multi-edition maps is that the geographic feature may expand or disappear so that the existing vector data might not provide accurate training samples (e.g., two segments of railroads in Figure 2(b) are not labeled by the contemporary vector data). If the system uses the contemporary vector data to label training samples in maps created in early years, the system might miss a part of the geographic feature or label incorrect geographic features. To solve this challenge, our system updates the vector data used for sample labeling using the extracted vector data from each map edition. The idea is that map content changes gradually between map editions. For example, the system needs to process the maps in Louisville, Colorado in 1942, 1950, and 1965. Figure 2 shows a part of railroads (the black lines with crosses) in Louisville in 1965, 1950, and 1942, respectively. Two segments of railroads are on the maps in 1957 and 1942, while they do not exist on the map in 1965. The system first processes the maps in 1965. Then the system updates the vector data based on the extraction results from the 1965 map. Next, the system uses the updated vector data to train a new CNN model to process the 1950 map. For the map in 1942, the system employs the vector data extracted from the 1950 map. The extracted vector data in 1950 can label the disappeared railroads in the 1942 map.

The difficulty of dealing with multi-scale maps is that the different scales may cause the vector data and the map misalignment problem (which is also common for maps of the same scale). Therefore the vector data might not be able to provide accurate training samples automatically. Figure 3 shows that the vector data overlays with the railroad at the same location but the different scales. The scale in Figure 3(a) is 1:24000, and the scale in Figure 3(b) is 1:31680. To solve this problem, we developed an algorithm that makes the vector data aligning with the geographic feature based on the location and geometry of the vector data using the intuition that the majority of vector data is nearby the geographic feature in a map.

(a) (b)

Figure 1: An example of the vector data (the red line in (b)) aligning with the railroads (the black line with crosses in (a))



(a) Year 1942 (b) Year 1950 (c) Year 1965

Figure 2: An example of the railroad (the black line with crosses) disappearance in Louisville, Colorado over time (the red line is the contemporary vector data)

(a) 1:24000 (b) 1:31680

Figure 3: An example of the railroad (the black line with crosses) in different scales (the red line is the vector data)

3 ARCHITECTURE OF OUR SYSTEM

Our system consists of three main components: recognition, semantic modeling, and post-processing.

The system uses CNN for feature recognition feeding with the training data generated automatically. The system uses existing vector data to find the location of the geographic feature of interest. The system automatically generates the training data by cropping the map using a certain sliding window along the vector data. Figure 1(b) shows an example of the vector data aligning with railroads.

We are building a generic semantic model to describe the contextual information of geographic features using existing geographic data and use the model to guide the post-processing processes. The model contains the locations of individual geographic feature (e.g., x-y coordinates of nodes and vertices), its type (e.g., railroads) and attributes (e.g., railroads width), geometry type (e.g., polylines), and geometric characteristics (e.g., a length of a railroad segment). Once the semantic model is built, the system will generate semantic descriptors by performing reasoning over the modeled data to infer semantic rules describing the spatial relationships between geographic features of the same and different types. The semantic descriptors will dictate changes in geometry or topological relations that are possible conflicts with those that are recognized as the geographic feature by mistake. In the post-processing, the system uses the semantic descriptors to detect the conflicts in order to improve the extraction results.

In the post-processing, the system vectorizes the recognition results and improve it using the semantic descriptors. The semantic descriptors detect the conflicts between extraction results and the semantic rules. For example, the semantic rule is that the railroads should be a long continuous line. The system discards the result, if it is a short segment.

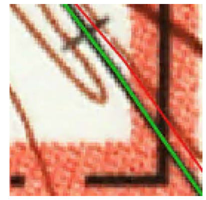


Figure 4: An example of the alignment algorithm results

4 PRELIMINARY RESULTS

We built a first version of the recognition system and tested our system on recognizing railroads the map in Bray, California in 2001. The semantic knowledge that the system used in the experiment is that the railroad should be a long continuous line without sharp curves. We use precision, recall, and F_1 as the metrics to evaluate the recognition results. We tested on two groups of vector data. One is the original vector data downloaded from the USGS website. The precision, recall, and F_1 are 29.30%, 98.38%, and 45.14% respectively. The other group is the aligned vector data processed by our alignment algorithm. Figure 4 shows an example of alignment algorithm performance. The red line is the original vector data. The green line is the aligned vector data processed by the algorithm. The black line with the cross underlying the green line is the railroad. The precision, recall, and F_1 for the aligned group are 57.98%, 92.36%, and 71.23% respectively. The aligned vector data group have a 100% improvement on the precision

5 DISCUSSION AND FUTURE WORK

The experiment shows that the fully automatic geographic feature recognition system is possible. Our system aligned the vector data for generating accurate training samples to improve the recognition performance. There are several other aspects that we can do to improve the performance. The first aspect is the CNN model. We are going to improve the recognition performance by building a more sophisticated CNN model. The second aspect is about the post-processing. The system only uses one semantic property of railroads. We plan to build more completed semantic descriptors. Besides improving the system, we are going to test on more maps.

ACKNOWLEDGEMENTS

This material is based on research supported in part by the National Science Foundation under Grant No. IIS 1563933 (to the Univesity of Colorado at Boulder) and IIS 1564164 (to the university of Southern California). We gratefully acknowledge the support of Microsoft Corporation with the Azure for Research Award and NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] Y.-Y. Chiang and C.A Knoblock. 2013. A general approach for extracting road vector data from raster maps. *IJDAR* (2013), 1–27.

[2] Y.-Y. Chiang and C.A Knoblock. 2015. Recognizing text in raster maps. *GeoInformatica* 19, 1 (2015), 1–27.

[3] E. Hancer, R. Samet, and D. Karaboga. 2014. A hybrid method to the reconstruction of contour lines from scanned topographic maps. In *ISIE*. IEEE, 930–933.

[4] T.C Henderson and T. Linton. 2009. Raster map image analysis. In *ICDAR'09*. IEEE, 376–380.

[5] A. Krizhevsky, I. Sutskever, and G. E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *ANIPS*. 1097–1105.

[6] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on CVPR workshops*. 806–813.

[7] K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).