Generating Named Road Vector Data from Raster Maps

Yao-Yi Chiang[1] and Craig A. Knoblock[2]

[1] University of Southern California,
Information Sciences Institute and Spatial Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292, USA
`yaoyichi@isi.edu`
[2] University of Southern California,
Department of Computer Science and Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292, USA
`knoblock@isi.edu`

Abstract. *Raster maps contain rich road information, such as the topology and names of roads, but this information is "locked" in images and inaccessible in a geographic information system (GIS). Previous approaches for road extraction from raster maps typically handle this problem as raster-to-vector conversion and hence the extracted road vector data are line segments without the knowledge of road names and where a road starts and ends. This paper presents a technique that builds on the results from our previous road vectorization and text recognition work to generate named road vector data from raster maps. This technique first segments road vectorization results using road intersections to determine the lines that represent individual roads in the map. Then the technique exploits spatial relationships between roads and recognized text labels to generate road names for individual road segments. We implemented this approach in our map processing system, called Strabo, and demonstrate that the system generates accurate named road vector data on example maps with 92.83% accuracy.*

Keywords: Raster map, road vectorization, text recognition, named road vector data, map labeling

1 Introduction

Cartographers have been making maps for centuries and road maps are one of the most used maps among all map types. Today we have access to a huge number of map collections in raster format from a variety of sources. For instance, the United States Geological Survey (USGS) has been mapping the United States since 1879. The USGS topographic maps at various time periods cover the entire country and contain informative geographic features, such as contour lines, buildings, and road lines. These raster maps are easily accessible compared to other geospatial data (e.g., road vector data) and present a unique opportunity for obtaining road information for the areas and time periods where and when road vector data do not otherwise exist. For example, we can generate named road vector data (road vector data that have a road-name attribute) from historical maps and build an accurate geocoder [Goldberg et al., 2009] or a gazetteer for spatiotemporal analysis of human-induced changes in the landscape.

Generating named road vector data from raster maps is challenging for a number of reasons. First, maps typically contain overlapping layers of geographic features, such as roads, contour lines, and text labels. Thus, the map content is usually highly complex and presents a difficult task for converting the road geometry in raster maps to vector format. Second, maps contain characters of various sizes constituting multi-oriented text labels, which cannot be recognized using classic optical character recognition (OCR) techniques. Finally, even after the road geometry is vectorized and text labels are recognized, there still exists the problem of labeling individual road lines with the recognized labels.

This paper presents an approach to generate named road vector data from raster maps while requiring minimal user effort. Figure 1 shows our overall approach, which integrates our previous map processing work (the interactive road vectorization [Chiang and Knoblock, 2011a] and text recognition techniques [Chiang, 2010; Chiang and Knoblock, 2011b]) and offers a new contribution: an automatic technique to identify individual road segments from the road vectorization results and then associate the recognized road labels with the road segments. This technique is the reverse engineering of cartographic-labeling methods [Agarwal et al., 1998; Doddi et al., 1997; Edmondson et al., 1996; Freeman, 2005]. The resulting named road vector data can be used in a geographic information system (GIS).

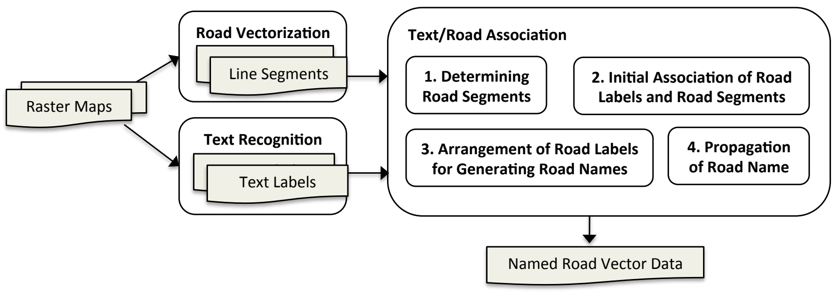


Fig. 1. The overall approach for generating named road vector data from heterogeneous raster maps

The remainder of this paper is organized as follows: Section 2 describes our previous map processing work on which the techniques in this paper built, Section 3 presents this paper's main contribution on associating road vector data with road labels, Section 4 reports on our experimental results, Section 5 discusses the related work, and Section 6 presents the discussion and future work.

2 Previous Work

This section briefly reviews our previous work on text recognition [Chiang, 2010; Chiang and Knoblock, 2011b] and road vectorization [Chiang and Knoblock, 2011a] from raster maps.

2.1 Text Recognition

In our previous work, we developed an interactive text recognition approach that requires only minimal user effort for processing heterogeneous raster maps [Chiang, 2010; Chiang and Knoblock, 2011b]. This approach first exploits a few examples of text areas for extracting text pixels and locating individual text strings. Figure 2 shows our user interface for labeling example text areas. Figure 3 shows an example map and the results where individual text strings are identified and shown in distinct colors (the color is only for explaining the idea). Once individual text strings are identified, we automatically detect the string orientations and rotate the strings to horizontal to then leverage conventional OCR software for recognizing the horizontal strings.

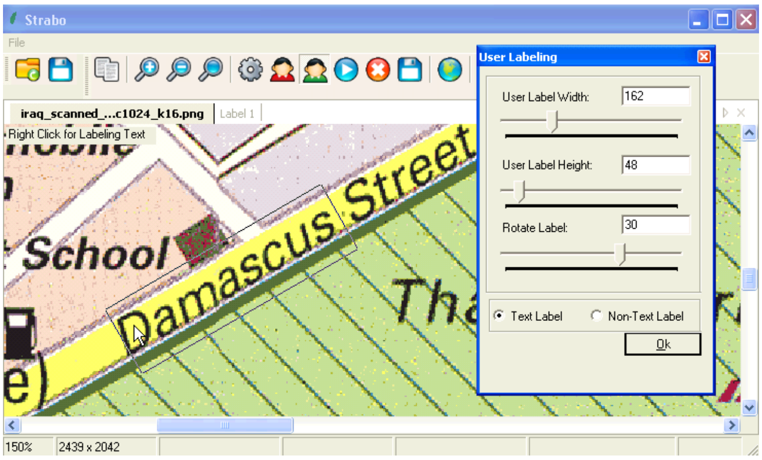


Fig. 2. Our user labeling interface for text recognition from raster maps

Fig. 3. Identify individual text labels

2.2 Road Vectorization

In our previous work, we developed an interactive road vectorization approach that requires minimal user effort to handle heterogeneous raster maps [Chiang and Knoblock, 2011a]. Similar to our text recognition approach, this road vectorization technique exploits a few examples of road areas to extract road pixels and generate road vector data.

To identify the road colors in a raster map for extracting the road pixels, our approach asks a user to first select a few example areas of roads. An example area of a road is a rectangle that is centered at a road intersection or a road segment. We exploit the fact that the road pixels in an example area of roads are a portion of one or more linear objects that are near the area center to determine the colors that represent roads in a map.

With the separated road layer (i.e., the set of extracted road pixels), we automatically detect the road width and format (i.e., single-line or double-line roads) and then dynamically generate parameters for applying the morphological operators (i.e., the dilation, erosion, and thinning operators) to extract and rebuild the road geometry (i.e., the centerline representation of the road network). The left image of Figure 4 shows an example map and the middle image shows the extracted road geometry, where the road lines near the intersections are distorted as a result of applying the morphological operators on thick lines. To extract accurate road vector data around the intersections, we detect the road intersections in the road geometry and label potential distortion areas around the intersections. Finally, we trace the thinned-line pixels outside the distortion areas to reconstruct the road intersections and generate the road vector data. The right image of Figure 4 shows the resulting road vector data where the road geometry around the intersections is accurate.

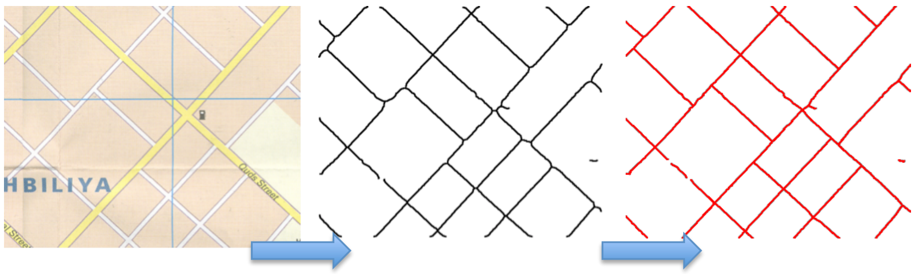


Fig. 4. Extract accurate road geometry and road vector data

3 Association of Road Vector Data and Road Labels

Our text/road association algorithm includes four major components. (i) The first component processes road vectorization results to generate individual road segments. Each road segment contains a set of line segments constituting the same road in a map. (ii) The second component assigns each road label to a road segment. (iii) For the road segments that are assigned with more than one road label, this component arranges the road labels to generate a road name using the relative positions between the assigned road labels and the road segment. (iv) Finally, the fourth component propagates the road names from road segments that have assigned road labels to the road segments that do not have assigned labels. In addition, if a road name is broken into several parts to label a long road in the input map, the separate parts are merged into a road name.

3.1 Determining Road Segments

The input to our road segmentation algorithm is the road vector data generated from our previous road vectorization work. The extracted road vector data contains a set of line segments, which are short, straight lines, without the knowledge of which line segments belong to each road segment in the map. Since road name changes commonly happen at road intersections, we use the locations of road intersections to group the input line segments into individual road segments – a road segment is a section of a road that is bounded by road endpoints or road intersections where more than two line segments meet. Figure 5 shows an example input and output of our road segmentation algorithm.

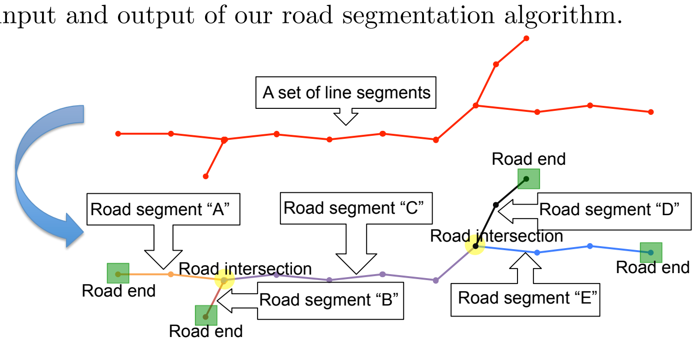
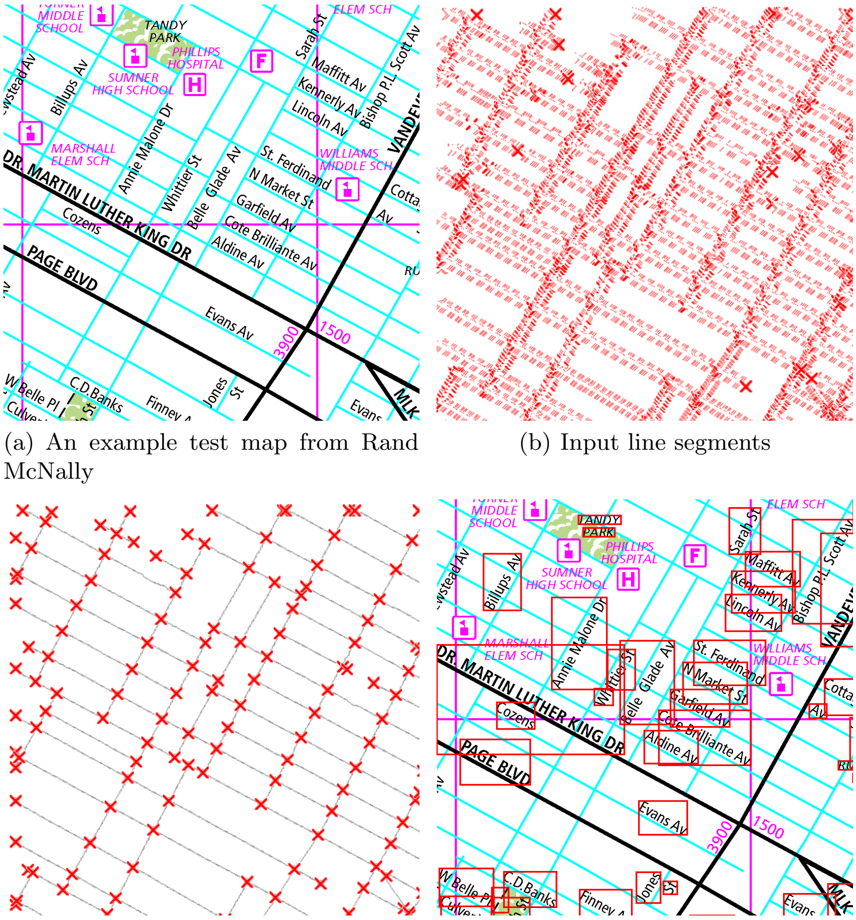


Fig. 5. Determining road segments from line segments based on road intersections

Our road segmentation algorithm first computes the connectivity of the endpoints of every line segments in the input road vector data. If an endpoint connects to only one other endpoint, the endpoint is a road end, namely a *RE* (the green squares in Figure 5). If an endpoint connects to more than two other endpoints, the endpoint is a road intersection, namely a *RI* (the yellow circles in Figure 5). If an endpoint is neither a *RE* nor a *RI*, the endpoint is a connecting point, namely a *CP*.

Once we have the connectivity of every endpoint of the input line segments, we iteratively process every input line segments until every line segment is assigned to a road segment. In one iteration, our algorithm starts from an unprocessed line segment and we first check the connectivity of its two endpoints. If the two endpoints are both classified as either a *RE* or *RI*, we assign the line segment as a road segment itself and then continue to process other line segments. If no or only one endpoint is classified as either a *RE* or *RI*, we search for the unprocessed line segments that connect to this line segment through the endpoints that are classified as a *CP*. We stop when the connected line segment we found contains a *RE* or *RI* or no line segment exists that is connected to this line segment. Figures 6(a), 6(b), and 6(c) show an example test map, the input road vector data, and the road segmentation results. The red crosses shows the endpoints of the line segments and the endpoints of road segments in Figures 6(b) and 6(c), respectively. The road segmentation results are then used in the next step with the recognized road labels to generate named road vector data. In an unusual case where the road name changes at non-road intersection locations, user input would be required to further separate the road segments.



(a) An example test map from Rand McNally

(b) Input line segments

(c) Output road segments

(d) The identified road labels in rectangular bounding boxes

Fig. 6. Example inputs and intermediate results for grouping road segments and determining the locations of road labels

3.2 Initial Association of Road Labels and Road Segments

Once we have the road segments, we start to assign each recognized road label to one of the road segments. Figure 6(d) shows the test map where the rectangles show the bounding boxes of the identified road labels. The recognized road labels, together with the identified road segments, are the input to this step.

Map labeling is a well investigated technique in both cartography [Edmondson et al., 1996] and computer science [Agarwal et al., 1998; Doddi et al., 1997;

Freeman, 2005]. In general, to label linear features in a map, a computer program or a cartographer places the labels in parallel to the corresponding linear features. The distance between a label and the corresponding feature should be smaller than the distance between the label to any other features of the same kind in the map. Therefore, to determine the correspondence between a road label and a road segment, we first assign every road label to the road segment that is the closest to the label and has the same orientation of the label.

To compute the distance between a road segment and a road label, we use the mass center of the road label to represent the position of this label. We calculate the distance between the mass center to each of the line segments in a road segment and use the shortest line-segment-to-mass-center distance as the distance between the road segment and the road label.

For a road label containing n character pixels, (x_i, y_i), the road label's mass center, (X_m, Y_m), is calculated as follows:

$$X_m = \frac{\sum_{i=1}^{n} x_i}{n}, Y_m = \frac{\sum_{i=1}^{n} y_i}{n} \tag{1}$$

To determine the parallelism between a road label and a road segment, we compare the orientation of the road segment with the orientation of the road label. The orientation of each road label is determined using our text recognition algorithm [Chiang and Knoblock, 2011b], and we compute the orientation of each road segment as follows: we first utilize the *Least-Squares Fitting* algorithm to find a straight line that best fits each road segment in the two dimensional space and then compute the orientation of the straight line. Assuming a linear function L for a set of points in a road segment, by minimizing the sum of the squares of the vertical offsets between the points and the line L, the *Least-Squares Fitting* algorithm finds the line L that most represents the road segment. For the target line function L as:

$$Y = m \times X + b, \tag{2}$$

the *Least-Squares Fitting* algorithm works as follows:

$$m = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \tag{3}$$

and

$$b = \frac{\sum y - m \sum x}{n} \tag{4}$$

With the line function of every road segment, we then derive the orientations of all road segments by applying the inverse trigonometric function, $ArcTan$, to the slope (m) of the road segments' line functions.

Because the orientations of the road labels and road segments are not estimated from the same type of data format (the road labels are a group of pixels and the road segments are vectors), to determine the parallelism between a road label and a road segment, we empirically define a buffer, B, as $10°$. If the difference between the orientations of a road label and a road segment is smaller than B, the road label and the road segment is determined to be in parallel.

For curved roads, if the road label is also curved along the curvature of the roads, the estimated orientation of the road label is similar to the road orientation determined using this approach. However, in the case where straight strings are used to label curved roads, the road label would not be assigned correctly and would need manual correction.

3.3 Arrangement of Road Labels for Generating Road Names

We can have multiple road labels assigned to a road segment if a road name is divided into more than one label in the map as shown in Figure 7. In this case, we need to determine the order of the assigned road labels for a road segment to then assemble the ordered road labels for generating a road name.

Fig. 7. More than one road labels can be assigned to a road segment

Given a road segment with multiple assigned road labels, for each of the road labels, we first determine which side of the road segment the road label appears in the map. This is determined using the cross product between the two *endpoints*, (X_s, Y_s) and (X_e, Y_e), of the road segment and the mass center, (X_m, Y_m), of the road label:

$$((X_e - X_s) \times (Y_m - Y_s)) - ((Y_e - Y_s) \times (X_m - X_s)) \tag{5}$$

The sign of the result from the cross product indicates which side the road label appears in the map. Once we have the relative position between the assigned road labels and the road segment, *we first rotate the road labels to the horizontal direction using the label orientation* and then check the relative position of the road labels and arrange the road labels as follows:

(i) If two road labels appear on the same side of the road segment, we order the labels using their X_m positions. This is because in English writing, a sequence of words is read from the left (a smaller X_m) to the right (a larger X_m).

(ii) If two road labels appear on different sides of the road segment, we order the road labels using their Y_m positions. Similarly, this is because in English writing, a sequence of words should be read from the top (a larger Y_m) toward the bottom (a smaller Y_m). For example, as shown in Figure 7, in the initial assignment, the road labels "Culver" and "Wy" are both assigned to the same road segment. After we rotate both road labels to the horizontal direction, the road label, "Culver", has a larger Y_m value among the two road labels, so it should be placed in front of "Wy". This case is also demonstrated using the road names "Jones St" and "Krum Av" in Figure 7.

Once we determine the order of the road labels for a road segment, we concatenate the ordered road labels to generate a merged road name.

3.4 Propagation of Road Names

Generally in computer map labeling and cartography map-making principles, not every road segment in a map is labeled with a road name because of the limited labeling space in the map and to avoid possible overlap of map labels. Therefore, repetitive road names are eliminated to improve the reading experience. For example, the "St. Louis Av" and the "University St" shown in Figure 8 are spread across more than one intersection, but the road names only appear once in the map. The green arrows indicate the possible start and end points of these roads that can be interpreted by a viewer. Moreover, words belong to a road name can be spread out to indicate the extent of a road, such as the "Greer Av" and " Dodier St" in Figure 8.

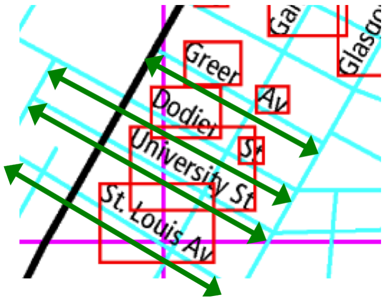


Fig. 8. Road labels do not repeat for each segment

As a result, after every road label is assigned to a road segment and multiple road labels that are assigned to a road segment are merged, we still need to assign road names to the road segments that do not have an assigned road label, and we need to merge the road labels that belong to the same road name but are assigned to more than one road segment. For example, we need to merge "Greer" and "Av" into "Greer Av" and then assign the merged road name to the corresponding road segments.

We start from a road segment, RS, that has an assigned road label, and we search for any other road segment that is connected to this road segment and has the same orientation. If a connected road segment, $NextRS$, has no assigned road label, we record that $NextRS$ has the same road name as RS. If $NextRS$ has assigned road labels, we order the assigned road labels of RS and $NextRS$ using the described method in the previous section. Then for the ordered road names, A followed by B, if B is a short string, we determine that the combination of A and B represents a road name and hence A and B should be merged. This is because if the last word in the sequence is a short string, this last word very often represents the road-type abbreviations (e.g., Av, St, Pl, Wy, and Dr). We define a short road label as a label of less than 5 characters since the longest abbreviations of the road types are "Blvd", which are 4 characters. This rule helps to merge two words into a complete road name.

The name propagation algorithm runs iteratively and records the number of road segments that have their road names assigned during each iteration. After an iteration, the algorithm checks if the number of road segments that have their road names assigned has increased. If the number stays the same, the algorithm stops since there are no road names that can be propagated. The results after this name propagation algorithm is a set of road segments, each labeled with a road name or an empty label indicating there is no road label in the map associated with this road segment.

4 Experiments

We have implemented the approach described in this paper in a system called Strabo. This section presents our experiments on Strabo for generating named road vector data from 6 raster maps of 2 map sources. The 2 sources are Rand McNally Maps (RM maps) and Afghanistan Information Management Services (AIMS maps).[3] Figure 6(a) shows an example RM map. The RM maps are designed for navigation purpose and contain very detailed road information. The RM maps represent common street maps that can be purchased in local gas stations and tourist stores. The AIMS maps contain only the information of major roads and are commonly used in urban planning.

We focus on evaluating the techniques for associating road names to the road vector data. The details of our road vectorization and text recognition results can be found in our previous work [Chiang, 2010; Chiang and Knoblock, 2011a]. To generate named road vector data from RM maps, we labeled 1 road area, 1 text area, and 1 non-text area. For AIMS maps, we labeled 1 road area and 1 text area. Based on these example areas, Strabo converted the road lines in the original maps to vector format, recognized the road labels, and generated named road vector data.

Strabo recognized 154 road labels and 892 road segments in the RM maps, and 15 road labels and 338 road segments in the AIMS maps. We manually verified each road segment using the test maps. Among the 892 road segments in RM maps, 866 road segments (97.09%) were correctly identified. Among the 338 road segments in AIMS maps, 327 road segments (96.75%) were correctly identified. The incorrect road segments have false road topology and/or geometry. Figure 9 shows an intersection where the road topology was incorrect due to the various road widths of the intersecting road lines. Sharp angles make the intersecting lines closer to each other and hence our road vectorization algorithm could not produce accurate geometry using the morphological operators.

To evaluate the overall performance for generating named road vector data, we define the accuracy as the length of correctly labeled road segments divided by the length of all identified road segments. A correctly labeled road segment is defined as follows: every line segment of a correctly labeled road segment represents a part or all of a road line that has the road name as the assigned name of the road segment. The accuracy for RM maps is 92.38% and for AIMS maps is 93.27%.

Figure 10 shows a portion of the extracted named road vector data displayed and labeled using Esri ArcMap. The yellow lines are the extracted road

[3] The information for obtaining the test maps and ground truth can be found on: `http://www.isi.edu/integration/data/maps/prj_map_extract_data.html`

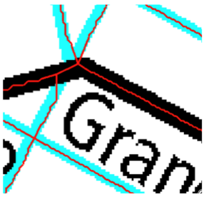


Fig. 9. Examples of incorrectly extracted road topology (red lines are the extracted road lines)

vector data and the red text with underlines are the assigned names.[4] From Figure 10, we can see that Strabo successfully propagated the road names to the corresponding road segments so that the road lines that are not labeled in the original map also had correctly assigned road names.

The majority of errors in our experimental results are due to the fact that some extracted road topology and/or geometry are incorrect. During the road vectorization process, the road topology could be incorrectly extracted due to the various road widths of the intersecting road lines. Because of this incorrect road topology, the road names of the connecting roads could not propagate through this intersection and resulted in falsely assigned road names or incomplete road names. Including a manual editing process for the results of the road vectorization and segmentation steps would reduce this type of error.

In addition, OCR could produce recognition errors. For example, in the test map, the string "BLVD" was recognized as "8LVD" and "Parnell St" was recognized as "Pamell St". If one or more characters of a road name was incorrectly recognized, the named road vector data results for the road segments associated with this road name were all considered to be incorrect.

Overall, Strabo generated accurate named road vector data: the average accuracy for the 6 maps from the 2 sources is 92.83%. To improve the results, we could have a user editor to process the extracted road vector data and recognized road labels for quality assurance so the text/road association algorithm could have more accurate input data.

5 Related Work

Map processing is an active area in both academic research and commercial software. However, to the best of our knowledge, the work presented in this paper is one of the first complete approaches to handling the problem of generating named road vector data from raster maps.

The most closely related work is a map computerizing system called MapScan [MapScan, 1998] from the United Nations Statistics Division. MapScan has the functionality for manually converting the linear features in raster maps into vector format and recognizing the text strings in raster maps. MapScan includes an extensive set of image processing tools (e.g., the morphological operators) and labeling functions for the user to manually computerize the raster maps, which requires intensive user input. For example, to recognize text strings using

[4] The map labeling algorithm of ArcMap did not label every road segments.

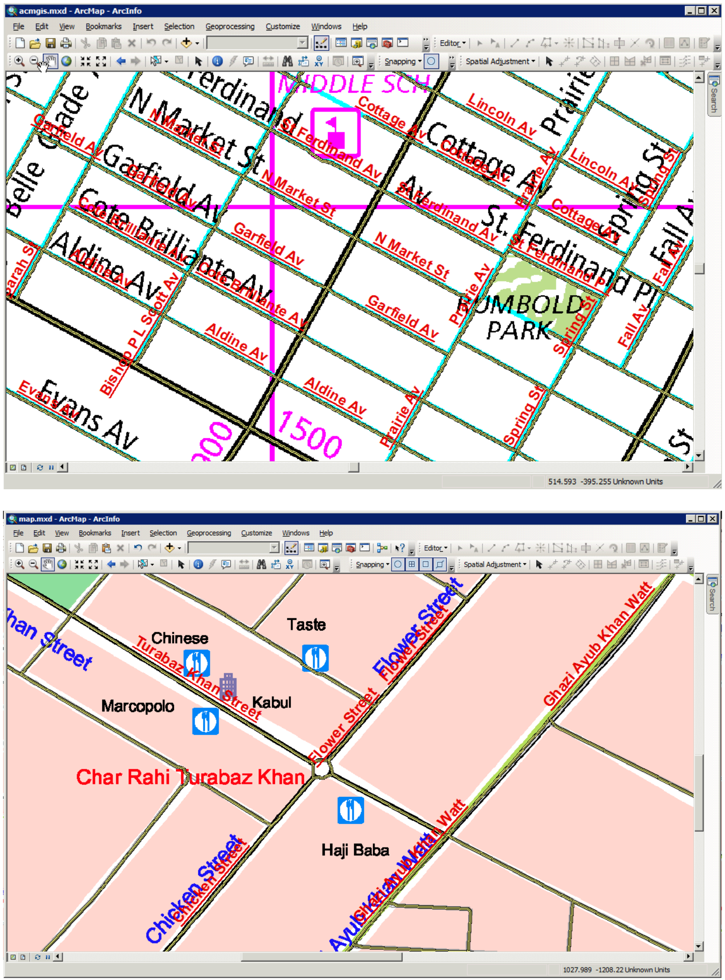


Fig. 10. The resulting named road vector data from RM and AIMS maps

MapScan, the user needs to label the areas of each text string and the string has to be in the horizontal direction. The association between the road names and extracted road vector data is achieved manually. In contrast, our approach requires only minimal user effort for recognizing road labels and extracting road vector data from raster maps, and further, we associate road names to road vector data automatically.

For text recognition from raster maps, Pouderoux et al. [2007] present a text recognition technique for raster maps. They identify text strings in a map by analyzing the geometric properties of individual connected components in the map and then rotate the identified strings horizontally for OCR. Roy et al. [2008] detect text lines from multi-oriented, straight or curved strings. Their algorithm handles curved strings by applying a fixed threshold on the connecting angle between the centers of three nearby characters. Their orientation detection method only allows a string to be classified into 1 of the 4 directions. In both [Pouderoux et al., 2007; Roy et al., 2008], their methods do not hold when the string characters have very different heights or widths. Moreover, these approaches handle specific types of road labels and do not work further to determine the association between the recognized road labels and the geographic features in raster maps.

For road vectorization from raster maps, Bin and Cheong [1998] extract road vector data from raster maps by identifying the medial lines of parallel road lines and then linking the medial lines. The linking of the medial lines requires various manually specified parameters for generating accurate results, such as the thresholds to group medial-line segments to produce accurate geometry of road intersections.

Itonaga et al. [2003] focus on non-scanned raster maps that contain only road and background areas. They exploit the geometric properties of roads (e.g., elongated polygons) to first label each map area as either a road or background area. Then they apply the thinning operator to extract a 1-pixel width road network from the identified road areas. The geometry distortions in the thinning results are then corrected by user-specified constraints, such as the maximum deviation between two intersecting lines.

In comparison to the approach in this paper, the techniques of Bin and Cheong [1998] and Itonaga et al. [2003] require significant user effort on parameter tuning. Moreover, their approaches do not determine where line segments compose a road segment in the vectorization result.

In addition to road vectorization research work, many commercial products offer the functionality for raster-to-vector conversion, such as Vextractor,[5] Raster-to-Vector,[6] and R2V from Able Software.[7] However, these commercial products do not have any text recognition capability, and hence do not work for generating named road vector data.

6 Discussion and Future Work

This paper presented a complete approach for generating named road vector data from raster maps. In particular, we presented an approach that automatically identifies individual road segments from road vectorization results and then associates recognized road labels with corresponding road segments. This approach, together with our previous road vectorization and text recognition work, allows a user to use only minimal effort for extracting named road vector data from raster maps. The resulting named road vector data are widely useful, such as for supporting a geocoder, building a gazetteer, and enriching available road information for spatial analysis in a GIS. In the future, we plan to test this work using raster maps with non-English labels. In addition, we plan to exploit the named road vector data generated from historical raster maps for spatiotemporal analysis.

[5] http://www.vextrasoft.com/vextractor.htm
[6] http://www.raster-vector.com/
[7] http://www.ablesw.com/r2v/

Bibliography

Agarwal, P. K., van Kreveld, M., and Suri, S. (1998). Label placement by maximum independent set in rectangles. *Computational Geometry*, 11(3-4):209 – 218.

Bin, D. and Cheong, W. K. (1998). A system for automatic extraction of road network from maps. In *Proceedings of the IEEE International Joint Symposia on Intelligence and Systems*, pages 359–366.

Chiang, Y.-Y. (2010). *Harvesting Geographic Features from Heterogeneous Raster Maps*. PhD thesis, University of Southern California.

Chiang, Y.-Y. and Knoblock, C. A. (2011a). A general approach for extracting road vector data from raster maps. *International Journal on Document Analysis and Recognition*, DOI: 10.1007/s10032-011-0177-1.

Chiang, Y.-Y. and Knoblock, C. A. (2011b). Recognition of multi-oriented, multi-sized, and curved text. In *Proceedings of the Eleventh International Conference on Document Analysis and Recognition*.

Doddi, S., Marathe, M. V., Mirzaian, A., Moret, B. M. E., and Zhu, B. (1997). Map labeling and its generalizations. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, pages 148–157.

Edmondson, S., Christensen, J., Marks, J., and Shieber, S. M. (1996). A general cartographic labelling algorithm. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 33(4):13–24.

Freeman, H. (2005). Automated cartographic text placement. *Pattern Recognition Letters*, 26:287–297.

Goldberg, D. W., Wilson, J. P., and Knoblock, C. A. (2009). Extracting geographic features from the internet to automatically build detailed regional gazetteers. *International Journal of Geographic Information Science*, 23(1):92–128.

Itonaga, W., Matsuda, I., Yoneyama, N., and Ito, S. (2003). Automatic extraction of road networks from map images. *Electronics and Communications in Japan (Part II: Electronics)*, 86(4):62–72.

MapScan (1998). *MapScan for Windows Software Package for Automatic Map Data Entry, User's Guide and Reference Manual*. Computer Software and Support for Population Activities, INT/96/P74, United Nations Statistics Division, New York, NY 10017, USA.

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

Roy, P. P., Pal, U., Llados, J., and Kimura, F. (2008). Multi-oriented English text line extraction using background and foreground information. *IAPR International Workshop on Document Analysis Systems*, pages 315–322.