Automatic Extraction of Road Intersections from Raster Maps

Yao-Yi Chiang, Craig A. Knoblock, and Ching-Chien Chen
University of Southern California
Department of Computer Science and Information Sciences Institute
Los Angeles, CA 90089-0781
[yaoyichi, knoblock] @isi.edu, chingchc@usc.edu

ABSTRACT

Numerous raster maps are available on the Internet, but the geographic coordinates of the maps are often unknown. In order to determine the precise location of a raster map, we exploit the fact that the layout of the road intersections within a certain area can be used to determine the map's location. In this paper, we describe an approach to automatically extract road intersections from arbitrary raster maps. Identifying the road intersections is difficult because raster maps typically contain multiple layers that represent roads, buildings, symbols, street names, or even contour lines, and the road layer needs to be automatically separated from other layers before road intersections can be extracted. We combine a variety of image processing and graphics recognition methods to automatically eliminate the other layers and then extract the road intersection points. During the extraction process, we determine the intersection connectivity (i.e., number of roads that meet at an intersection) and the road orientations. This information helps in matching the extracted intersections with intersections from known sources (e.g., vector data or satellite imagery). For the problem of road intersection extraction, we applied the techniques to a set of 48 randomly selected raster maps from various sources and achieved over 90% precision with over 75% recall. These results are sufficient to automatically align raster maps with other geographic sources, which makes it possible to determine the precise coverage and scale of the raster maps.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial Databases and GIS*

General Terms

Algorithms, Design

Keywords

Raster map, road extraction, road intersection, imagery, conflation

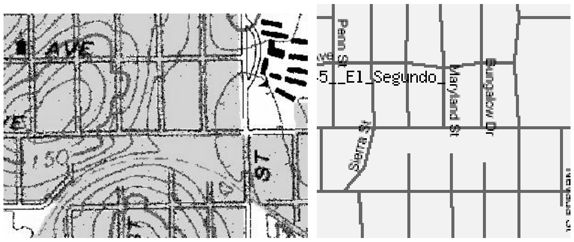
1. INTRODUCTION

Due to the popularity of Geographic Information System (GIS) and high quality scanners, we can now obtain more and more raster maps from various sources on the Internet, such as digitally scanned USGS Topographic Maps on Microsoft Terraserver,[1] or computer generated TIGER/Line Maps from U.S Census Bureau,[2] ESRI Maps,[3] Yahoo Maps,[4] MapQuest Maps,[5] etc. To utilize these raster maps, we need to know the geospatial coordinates of the maps. The fact is, however, only a few map sources provide this information. In addition, among the sources that provide the geospatial coordinates of the maps, only a few embed the information in the raster maps using the geo-tiff format, while others include it on the companion web pages or in separate files, which may result in the loss of this information if the raster map and the geospatial coordinates are provided separately.

If the map sources do not provide geospatial coordinates or the information is missing, we need an alternative way to identify the coordinate of raster maps. Considering the fact that the road layers are commonly used on various raster maps and the road networks are usually distinguishable from each other, we can use the road intersection points as the "fingerprint" of the raster maps. By matching a set of road intersection points of a raster map covering an unknown area to another set of road intersection points for which the geospatial coordinates are known, we can identify the coverage of the unknown raster maps.

In our previous work [5], we described an automatic and accurate map conflation method to integrate raster maps with orthoimagery. We combine the information on raster maps with accurate, up-to-date imagery by matching the corresponding features (i.e., road intersection point sets) between raster maps and imagery. In that work, we developed a simple approach to detect intersections from simpler raster maps, and we only used the positions of the road intersection points in the matching process during the conflation. In this paper, we present a more general approach to handle diverse and more complicated maps (e.g., USGS Topographic Maps, Thomas Brother Maps). We achieve higher precision/recall and also effectively compute the intersection connectivity and the road orientations to help a conflation system to prune the search space during the matching process.

[1] http://terraserver-usa.com/
[2] http://tiger.census.gov/cgi-bin/mapsurfer
[3] http://arcweb.esri.com/sc/viewer/index.html
[4] http://map.yahoo.com
[5] http://www.mapquest.com



(a) USGS Topographic Map (b) TIGER/Line Map

Figure 1: The raster maps that use double lines and single lines to represent road layers (El Segundo, CA)

Raster maps in different scales are composed from multiple layers which typically contain information about roads, buildings, symbols, and characters. In some cases, usually for computer generated raster maps, road layers can be separated by extracting pixels with user specified colors or colors learned from the legend information. There are still a huge number of raster maps, however, for which we cannot separate road layers using specified color thresholds (e.g., low-quality scanned maps, USGS Topographic Maps). To overcome this problem, we first utilize an automatic segmentation algorithm to remove background pixels based on the difference in the luminosity level. After we obtain the foreground pixels, which contain the entire information layer of the original raster map, we separate the road layer from other layers to extract the road intersection points.

Road layers are usually presented in single-line or double-line format depending on map sources as shown in Figure 1. The double-line format provides us more information to extract the road layers than the single-line format, which is important if the input raster map has other layers that contain mostly linear structures, such as grid lines, rivers or contour lines. We automatically examine the format of the road layer on the input raster map, and detect the road width if the road layer is in double-line format to trace the parallel pattern of the road lines. Then we apply text/graphics separation algorithms with morphological operators to remove noise and rebuild the road layer.

With the extracted road layer, we detect salient points as the road intersection candidates based on the variation of luminosity level around each road layer pixel. The connectivity of each salient point is then computed by checking the neighborhood linear structures on the road layer (i.e., road lines) to determine if it is an actual road intersection point. We also compute the orientation of the roads intersecting at each road intersection point as a by-product.

The remainder of this paper is organized as follows. Section 2 describes our approach to extract road intersections. Section 3 reports on our experimental results. Section 4 discusses the related work and Section 5 presents the conclusion and future work.

2. AUTOMATIC ROAD INTERSECTION DETECTION

The overall approach we are taking in this paper is shown in Figure 2. The input can be any raster map regardless of resolution from various sources and without any prior knowledge such as color of layers, vector data, legend types or gazetteer data [6]. The outputs are the positions of road intersection points as well as the connectivity and the orientation of each intersected road.

Raster maps usually contain many objects, such as characters, buildings, streets, rivers or even contour lines in topographic maps, and it is important that we can distinguish between these objects and the background. We classify the input raster maps into two major categories depending on the way they were generated. The first category includes computer generated raster maps from vector data, such as the TIGER/Line Maps, and the other includes scanned raster maps, such as USGS Topographic Maps.

Computer generated raster map sources usually use different colors to represent different information layers, especially road layers. Thus road layers can be extracted using a specified color threshold. Different raster map sources, however, require different color thresholds, and even raster maps from the same source may have different color thresholds for different scales. The scale or the source of the input raster maps are unknown in our automatic approach, and we are not able to use only a color threshold to separate the road layers. On the other hand, scanned raster maps suffer from quantization errors resulting from the manual scan process [11] and the color of each layer may vary from tile to tile. For example, in the USGS Topographic Maps some roads are composed of brown, white and black pixels and others are composed of pink, brown, and black pixels.

Since the color threshold is not a reliable property to extract the road layer, we use the differences in luminosity level to remove background pixels in the first module, and use the geometry properties of road lines to separate road layers from others among foreground pixels in the second module. The last module detects salient points and determines which salient points should be identified as road intersection points by counting the connectivity of each salient point along with the orientation of each intersected road.

2.1 Automatic Segmentation

In order to automatically separate the foreground without introducing additional noise, we use a common technique called segmentation. We first discard color information by converting the original input raster maps to 8 bit grayscale with 256 color levels. Then we use the luminosity as a clue to automatically generate a threshold by the Triangle method proposed by Zack et al.[17]. The segmentation uses the threshold to segment the foreground pixels and background pixels. The grayscale and binary images are shown in Figure 3.a and Figure 3.b.

2.2 Pre-Processing - Extracting Road Layers

This module receives the binary raster map that contains multiple information layers as input and outputs the road layer. Road layers on raster maps typically have two distinguishable geometric properties from other layers:

1. Road lines are straight within a small distance (i.e., several meters in a street block).

2. Unlike label layers or building layers, which could have many small connected objects; road lines are connected to each other as road networks and road layers usually have few connected objects or even only one huge connected object - the whole road layer.

Some map sources use double lines to represent roads, like Yahoo Maps, ESRI Maps, or USGS Topographic Maps, while others use single lines. Double-line format is com-

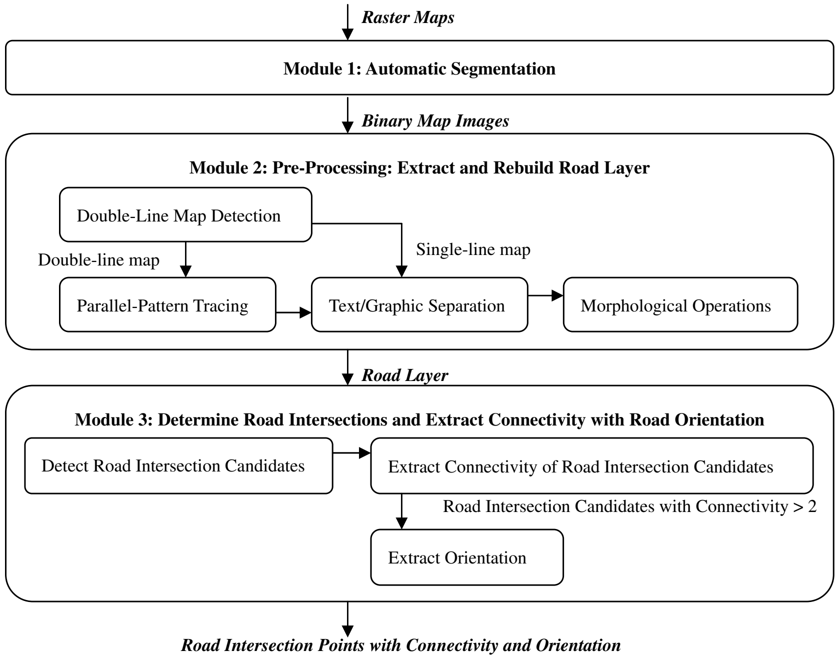
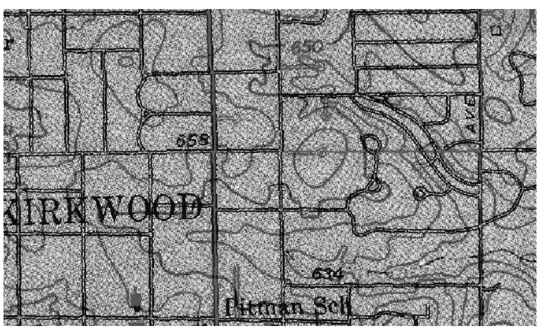
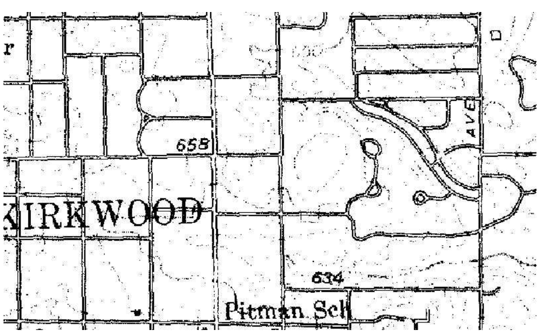


Figure 2: The overall approach to extract road intersections



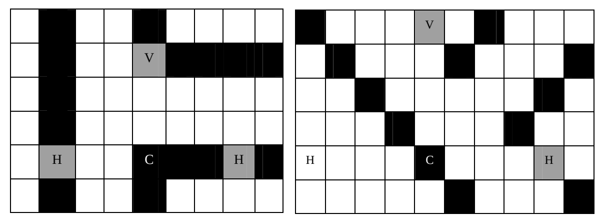
(a) Grayscale map



(b) Binary map

Figure 3: Raster maps before and after automatic segmentation (USGS Topographic Map, St. Louis, MO)

monly used when the resolution is high or the maps contain other linear objects such as contour lines. After automatically checking the format of the road layer, we use parallel-pattern tracing to eliminate linear structures other than road lines if the map uses double-line format. The text/graphics separation program then removes small connected objects with part of the road lines that touch the removed objects, and the morphological operators reconnect the broken road lines.



(a) RW = 3 (b) RW = 4

Figure 4: Double-line format checking and parallel-pattern tracing (C is the target foreground pixel. V is the pixel at the vertical direction and H is at the horizontal direction. Black cells are foreground pixels.)

2.2.1 *Double-Line Format Checking and Parallel-Pattern Tracing*

To determine whether a target foreground pixel is on a double-line road layer with a road width of RW pixels, we search for the corresponding foreground pixels at a distance of RW in horizontal and vertical directions. If the target pixel is on a horizontal or vertical road line, we can find two foreground pixels along the orientation of the road line within a distance of RW and at least another foreground pixel on the corresponding parallel road line in a distance of RW, as shown in Figure 4.a. If the orientation of the road line is neither horizontal nor vertical, we can find one foreground pixel on each of the horizontal and vertical direction on the corresponding parallel road lines at a distance of RW, as shown in Figure 4.b.

There are some exceptions, however, as shown in Figure 5; foreground pixels from 1 to 8 are the example pixels which have the above properties (gray pixels are the corresponding pixels in horizontal/vertical direction or on the corresponding parallel road line of these pixels) and foreground pixels from A to D are the example pixels that belong to the double road line layer but do not have the above properties. After the parallel-pattern tracing, pixels A to D will be removed resulting in small gaps between line segments. These gaps

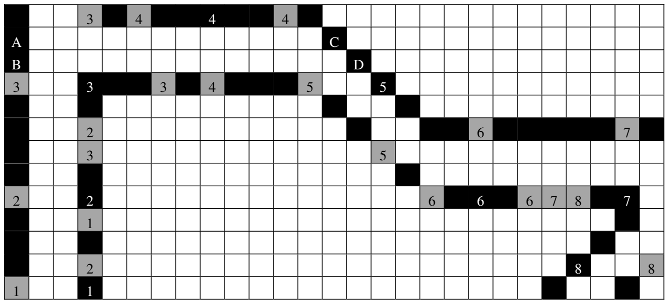
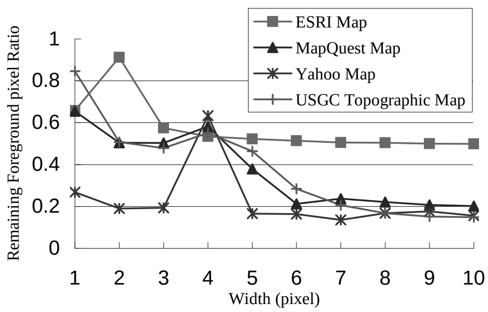
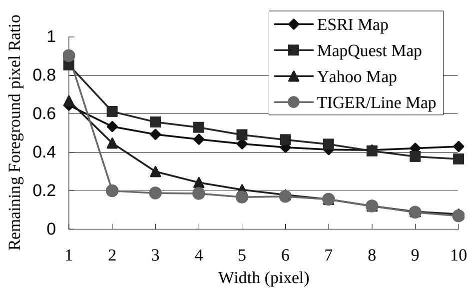


Figure 5: The exceptions in double-line format checking and parallel-pattern tracing (white cells are background pixels)



(a) The raster map sources with double-line format road layers

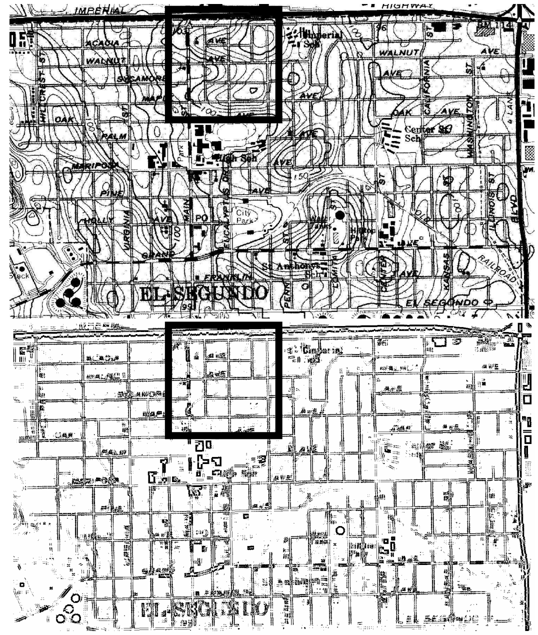


(b) The raster map sources with single-line format road layers

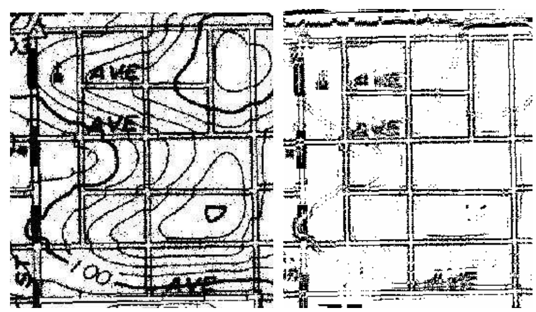
Figure 6: Double-line format checking

will be fixed later using the morphological operators. Although we use single pixel-wide road lines in Figure 4 and 5 for simplification, road lines which are multiple pixels wide are also suitable for parallel-pattern tracing. To utilize the parallel-pattern tracing component, we need to know the format of the input road layer and the road width (RW). We check the road layer format by applying parallel-pattern tracing on the input raster maps varying the road width from 0 to 10 pixels and remove foreground pixels which do not have the properties to be a road line pixel for a given road width. Then we compute the ratio of the remaining foreground pixels divided by the original foreground pixels for each road width as shown in Figure 6.

At the beginning of double-line format checking process, we set the road width as 0 pixel and no foreground pixel is removed. After increasing road width, the ratio starts to decline. This is because foreground pixels tend to be near each other, and it is easier to find corresponding pixels even if the road width is not correct or it is not a double-line map when the given road width is small. If the input raster map



(a) Full-size view with interest area highlighted by the black rectangle



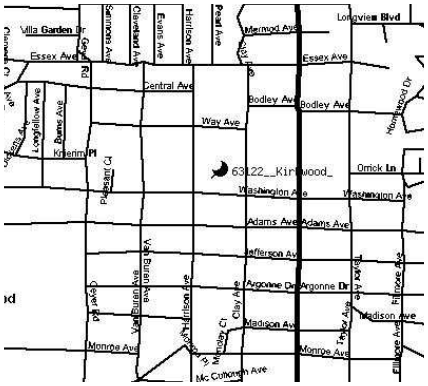
(b) Detail view

Figure 7: USGS Topographic Map before and after parallel-pattern tracing (El Segundo, CA)

has a double-line road layer with the correct road width, there is a peak on the line of the chart in Figure 6.a because the majority of foreground pixels on the double-line road layer have corresponding foreground pixels. ESRI Maps, MapQuest Maps, Yahoo Maps in high resolution and all of the USGS Topographic Maps are double-line maps that have a peak on the line as shown in Figure 6.a. ESRI Maps and MapQuest Maps, which are not high resolution, and all of the TIGER/Line Maps are all single-line maps, which do not have any peak as shown in Figure 6.b. Using this method, we can detect double-line maps automatically and also obtain the road width by searching for the peak. For example, from Figure 6.a, we know the USGS Topographic Map is a double-line map with road width equal to 4 pixels. Hence, we apply the parallel tracing algorithm setting RW to 4 pixels. The resulting image of this step is shown in Figure 7 and the remaining pixels are mainly road lines with some broken characters. The contour lines and other linear structures are all removed.

2.2.2 Text/Graphics separation

After we use the parallel-pattern tracing algorithm to eliminate other layers which contain linear structures, the re-



(a) Binary TIGER/Line map

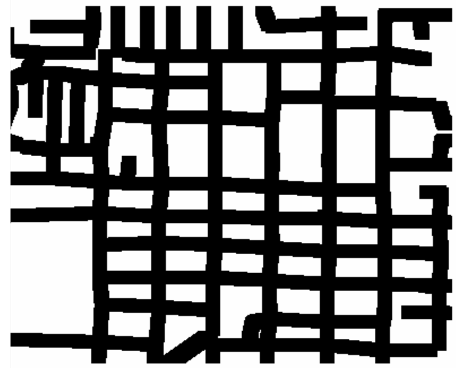
(b) After text/graphics separation

Figure 8: TIGER/Line map before and after text/graphics separation (St. Louis, MO)

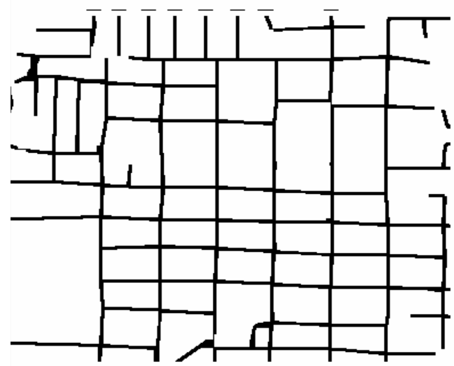
maining major sources of noise are the small connected objects, e.g. buildings, symbols, characters, etc. The small connected objects tend to be near each other on the raster maps, such as characters that are close to each other to form a string, and buildings that are close to each other on a street block. The text/graphics separation algorithms in pattern recognition [2, 3, 7, 9, 15, 16] are very suitable for grouping these small connected objects. These text/graphics separation algorithms start by identifying every small connected foreground object, and then use various algorithms to search neighborhood objects in order to build an object group [15]. We apply the algorithm described in Cao et al.[3] and the result is shown in Figure 8. The broken road lines are inevitable after the removal of those objects touching the lines, and we can reconnect them later using the morphological operators described next.

2.2.3 *Morphological Operators: Generalized Dilation, Generalized Erosion and Thinning*

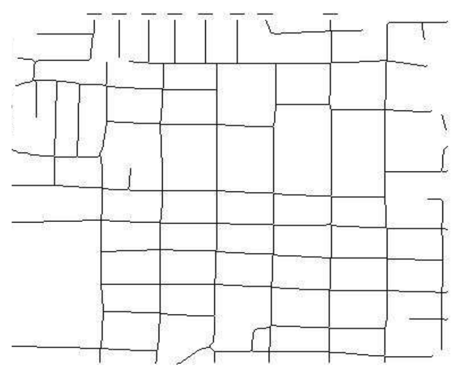
Morphological operators are implemented using hit-or-miss transformations [10], and the hit-or-miss transformation is performed in our approach as follows: We use 3-by-3 binary masks to scan over the input binary images. If the masks match the underneath pixels, it is a "hit," and if the masks does not match the underneath pixels, it is a "miss." Each of the operators uses different masks to perform hit-or-miss transformations and performs different actions as a result of a "hit" or "miss." We briefly describe each operator in the following paragraphs and the resulting images after each operator are shown in Figure 9.



(a) After generalized dilation operator



(b) After generalized erosion operator

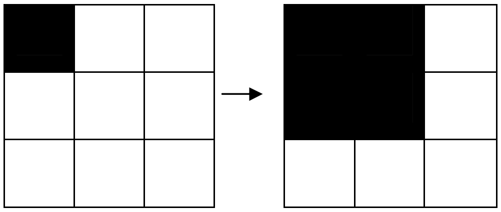


(c) After thinning operator

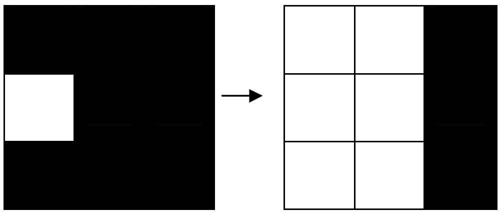
Figure 9: The resulting images from morphological operators. The input is shown in Figure 8.b

The effect of a generalized dilation operator is expanding the region of foreground pixels [10]. We use it to thicken the road lines and reconnect the neighbor pixels. As shown in Figure 10.a, if a background pixel has a foreground pixel in any of its eight neighbor pixels (i.e., a "hit"), it will be filled up as a foreground pixel (i.e., the action resulting from the "hit"). The resulting image after performing three iterations of the generalized dilation operator on Figure 8.b is shown in Figure 9.a. The number of iterations determines the maximum size of gaps we want to fix. The gaps smaller than 6 pixels are now reconnected and road lines are thicker.

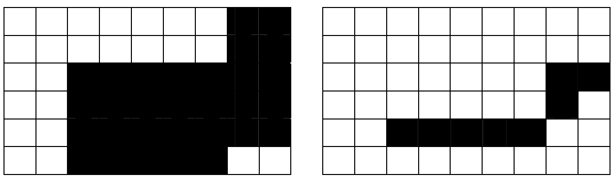
The idea of a generalized erosion operator is to reduce the region of foreground pixels [10]. We use it to thin the road lines and maintain the orientation similar to the original orientation prior to applying the morphological operators. If a foreground pixel has a background pixel in any of its eight neighbor pixels (i.e., a "hit"), it will be erased as a back-



(a) The generalized dilation operator



(b) The generalized erosion operator



(c) The thinning operator

Figure 10: The morphological operators (black cells are foreground pixels)

ground pixel (i.e., the action resulting from the "hit") as shown in Figure 10.b. The resulting image after performing two iterations of the generalized erosion operator on Figure 9.a is shown in Figure 9.b. The road lines are thinner and the orientation is similar to the original.

After applying the generalized dilation and erosion operators, we have road layers composed from road lines with different width, but we need the road lines to have exactly one pixel width to detect salient points and the connectivity in the next module. The thinning operator can produce the one pixel width results as shown in Figure 10.c. The idea of using the generalized erosion operator before the thinning operator is because the generalized erosion operator has the opposite effect to the generalized dilation operator, which can prevent the orientation of road lines from being distorted by the thinning operator. The thinning operators are conditional erosion operators which have an extra confirmation step. After we mark all possible foreground pixels to be converted to background pixels in the first step, the confirmation step utilizes the conditional masks to determine which pixel among the candidate pixels should be converted to background pixels to ensure the conversion will not compromise the basic structure of the original objects. The resulting image with the extracted road layers after applying the thinning operator on Figure 9.b is shown in Figure 9.c.

2.3 Detection of Road Intersection Candidates

After eliminating the layers other than the road layer, we need to locate possible road intersection points. A salient point is a point at which more than one line segment meets with different tangents, which is the basic requirement of a road intersection point. Among the image processing operators, the interest operator is most suitable to detect the salient points such as corners or intersections on the input

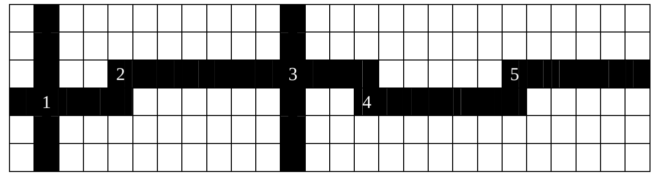
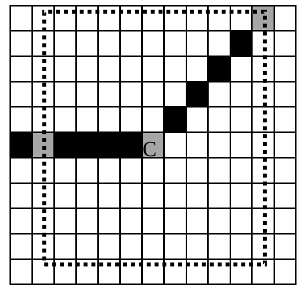
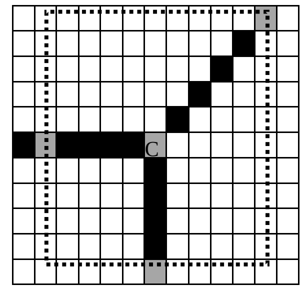


Figure 11: The salient points (black cells are foreground pixels)



(a) 2-line connectivity, not an intersection point



(b) 3-line connectivity, an intersection point

Figure 12: Use 11-by-11 rectangles to get connectivity for road intersection filtering. C is the road intersection candidate detected by the interest operator (black cells are foreground pixels).

road layer. We use the interest operator proposed by Shi and Tomasi [13] and implemented in OpenCV[6] to find the salient points as the road intersection candidates.

The interest operator checks the color variation around every foreground pixel to identify salient points, and it assigns a quality value to each salient point. If one salient point lies within the predefined radius R of some salient points with higher quality value, it will be discarded. As shown in Figure 11, pixel 1 to pixel 5 are all salient points, with the radius R of 5 pixels. Salient point 2 is too close to salient point 1, which has a higher quality value. We discard salient point 2, while salient point 1 will become a road intersection candidate. We also discard salient point 4 because it lies within the 5 pixels radius of salient point 3. Salient point 5 is considered as a road intersection point candidate, however, since it does not lie within any other salient points with higher quality value. These road intersection candidates are then passed to the next module for the determination of actual road intersections.

2.4 Filtering Intersections, Extracting Intersection Connectivity and Road Orientation

The definition of intersection connectivity is the number of line segments intersecting at an intersection point. Every road intersection point should be crossed by more than one road, which is more than two line segments. The connectivity is the main criteria to distinguish road intersection points from salient points.

We assume roads on raster maps are straight within a small distance (i.e., several meters within a street block). For each salient point detected by the interest operator (i.e., GoodFeaturesToTrack function in OpenCV), we draw a rectangle around it as shown in Figure 12. The size of the rect-

[6] http://sourceforge.net/projects/opencvlibrary

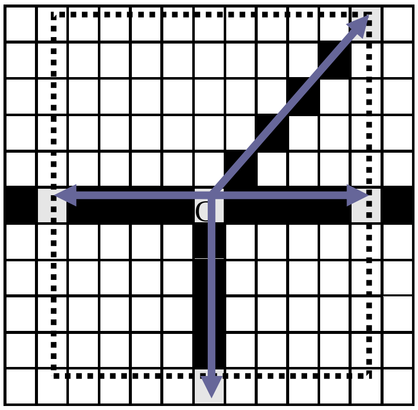


Figure 13: Construct lines to compute orientation

angle is based on the maximum length in our assumption that the road lines are straight. In Figure 12, we use an 11-by-11 rectangle on the raster map with resolution 2m/pixel, which means we assume the road lines are straight within 5 pixels (e.g., on the horizontal direction, a line of length 11 pixels is divided as 5 pixels to the left, one center pixel and 5 pixels to the right), 10 meters. Although the rectangle size can vary with various raster maps of different resolutions, we use a small rectangle to assure even with the raster maps of lower resolution, the assumption that road lines within the rectangle are straight is still tenable.

The connectivity of the salient point is the number of foreground pixels that intersect with this rectangle since the road lines are all single pixel width. If the connectivity is less than three, we discard the point; otherwise it is identified as a road intersection point. Subsequently, we link the salient point to the intersected foreground pixels on the rectangle to compute the slope (i.e., orientation) of the road lines as shown in Figure 13.

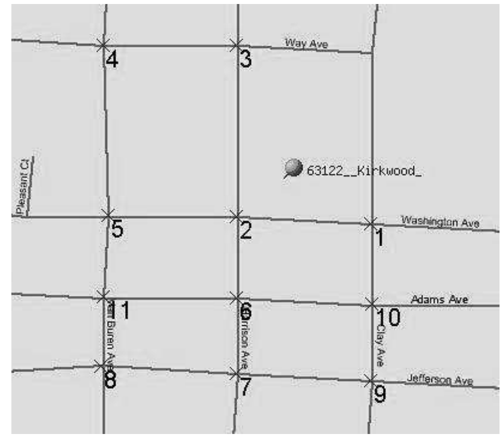
In this module, we skip the step to trace the pixels between the center pixel and the intersected pixels on the rectangle. This could introduce errors if the intersected pixels are from other road lines which do not intersect on the center pixel or the road lines within the rectangle are not straight. This usually happens in low-resolution maps, however, in the general case, the rectangle is much smaller than the size of a street block, and it is unlikely to have other road lines intersect or have non-straight road lines. Moreover, we save significant computation time by avoiding the tracing of every pixel between the center and the rectangle box.

3. EXPERIMENTS

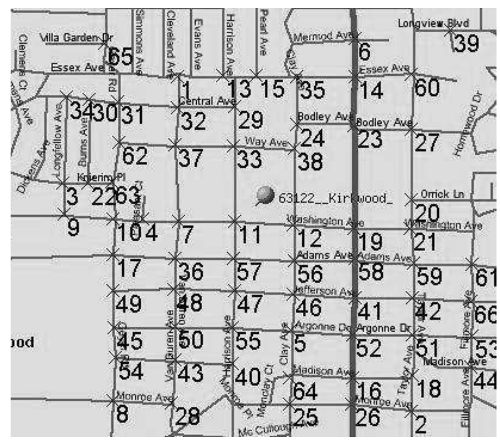
We experimented with six sources of raster maps, ESRI Map, MapQuest Map, Yahoo Map, TIGER/Line Map, USGS Topographic Map and Thomas Brothers Los Angeles 2003 Map, with different map scales as shown in Table 1. USGS Topographic Map and Thomas Brothers Map are scanned maps while the others are computer generated from vector data. These raster maps are randomly selected within the areas covering El Segundo, CA and St. Louis, MO.

3.1 Experimental Setup

Since we assume that we do not have any information about the input maps, we use a set of default thresholds for all the input raster maps. The size of small connected objects to be removed in text/graphics separation program



(a) 1.85m/pixel



(b) 4.17m/pixel

Figure 14: Road intersection extraction (TIGER/Line Map, St. Louis, MO)

is set to 20-by-20 pixels, which means any object smaller than this size will be removed. The number of iterations for the generalized dilation operator is 3 and for the generalized erosion operator is 2 (i.e., a gap smaller than 6 pixels can be fixed). In the filtering intersection and extracting connectivity and orientation module, we used a 21-by-21 pixels rectangle box (10 pixels to the left, 10 pixels to the right plus the center pixel).

These thresholds are based on practical experiences and may not have the best results for all raster map sources, but the results are good enough to generate a set of road intersection points to identify the raster maps [4]. We can optimize them for one particular source to get the best precision/recall if we know the source of the input raster maps.

3.2 Experimental Results

The resulting images from our experiments are shown in Figure 14 and Figure 15. In these figures, an “X” means one road intersection point extracted by our system, and the number next to each “X” is shown for the users to examine the matched result after the conflation. The statistics are in Table 1 and Table 2. The precision is defined as the number of correctly extracted road intersection points divided by the number of extracted road intersection points. The recall is defined as the number of correctly extracted road inter-

Table 1: Experimental results, P/R/A (i.e., Precision/Recall/Positional Accuracy), with respect to raster map source, resolution and type of the road layer (El Segundo, CA and St. Louis, MO)

Map Source	Map Type	P/R/A(pixel) by Source	Type of the Road Layers	P/R/A(pixel) by Type of the Road Layers	Resolution (m/pixel)	P/R by Resolution (m/pixel)	Number of Tested Maps
ESRI Map	Computer generated	0.96/0.64/0.43	Double line	0.96/0.64/0.43	N/A*	0.96/0.64	10
MapQuest Map	Computer generated	0.90/0.61/0.84	Double line	1.00/0.88/0.57	2.00	1.00/0.85	1
					2.17	1.00/0.89	2
			Single line	0.87/0.52/0.93	4.84	0.92/0.75	3
					5.17	0.95/0.65	3
					11.11	0.80/0.20	2
					11.67	0.59/0.09	1
TIGER/Line Map	Computer generated	0.94/0.74/0.57	Single line	0.94/0.74/0.57	1.85	1.00/1.00	1
					2.90	0.98/0.72	1
					3.82	0.92/0.67	4
					4.17	0.97/0.85	2
					7.65	0.84/0.38	1
					7.99	0.95/0.84	1
USGS Topographic Map	Scanned	0.84/0.74/0.80	Double line	0.84/0.74/0.8	2.00	0.84/0.74	10
Yahoo Map	Computer generated	0.86/0.64/0.11	Double line	0.96/0.81/0.07	1.20	1.00/0.96	1
					1.22	0.83/0.90	2
			Single line	0.34/0.07/0.13	4.26	0.99/0.76	7
					14.08	0.34/0.07	2
Thomas Brothers Map	Scanned	0.94/0.66/0.01	Single line	0.94/0.66/0.01	N/A**	0.94/0.66	2

* We deliberately chose the ERSI map service that does not provide the resolution.

** We randomly selected two scanned Thomas Brothers Map without the knowledge of the scanned resolution

Table 2: Experimental results with respect to resolution (El Segundo, CA and St. Louis, MO)

Resolution	Precision	Recall
Higher then 7m/pixel (48 maps)	0.92	0.77
Lower then 7m/pixel (8 maps)	0.66	0.27

section points divided by the number of road intersections on the raster map. The positional accuracy is defined as the distance in pixels between the correctly extracted road intersection points and the corresponding actual road intersections. Correctly extracted road intersection points are defined as follows: if we can find a road intersection on the original raster map within a 5 pixel radius of the extracted road intersection point, it is considered a correctly extracted road intersection point. Road intersections on the original maps are defined as the intersection points of any two roads if it is a single-line map or the intersection areas where any two roads intersect if it is a double-line map.

The low-resolution maps (i.e., resolutions lower than 7m/pixel) have below average precision and low recall as shown in Table 2. This is because the characters and symbols touch the lines more frequently as shown in Figure 16. In the preprocessing step, we use text/graphics separation program to remove the characters and labels, and it will remove most of the road lines in a low-resolution map. Also, the size of street blocks on the low-resolution map is usually smaller than the window size we use in the intersection filter, which leads to inaccurate identification of road orientations.

Except for the low-resolution maps in our experiments, the USGS Topographic Maps have the lowest precision and recall. This is because a USGS Topographic Map contains

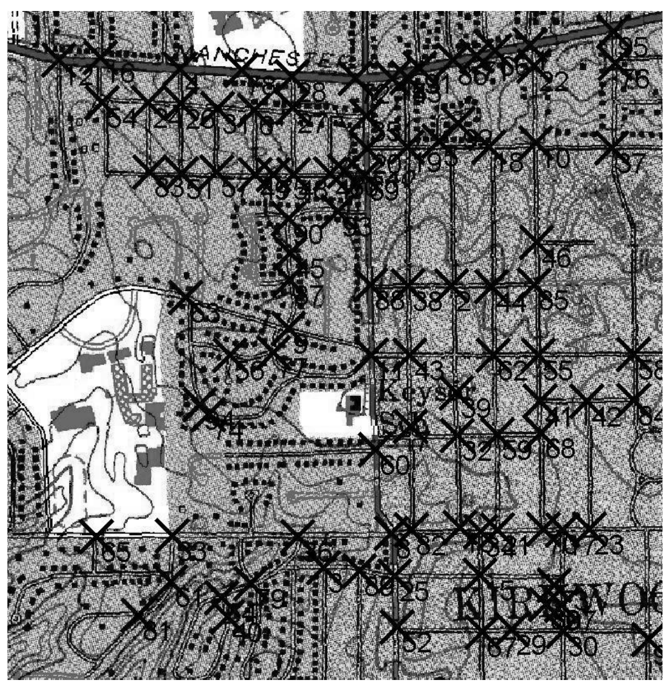


Figure 15: Road intersection extraction (USGS Topographic Map, 2m/pixel, St. Louis, MO)

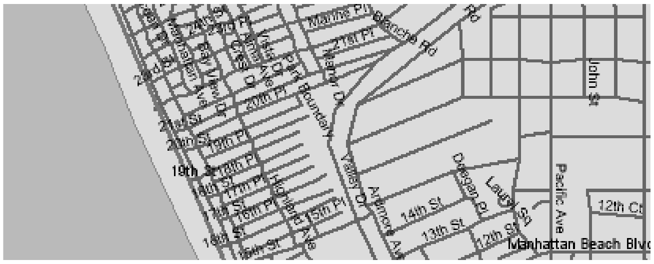


Figure 16: A low-resolution raster map (TIGER/Line Map,7.99m/pixel)

more information layers than other map sources and the quality of scanned maps is not as good as computer generated maps.

We also report the positional accuracy because the morphological operators may cause the extracted road layers to shift from the original position. The average positional accuracy is lower than 1 pixel in our experiments. This means the average distance between the intersection points we found and the actual intersection points are less than 1 pixel. Thus providing a good set of features for a conflation system to precisely align the raster maps with other sources.

The computation time mainly depends on how many foreground pixels are in the raster maps. Raster maps which contain more information need more time than others. USGS Topographic Maps are the most informative raster maps in our experiments; it took less than one minute to extract the road intersections from an 800 x 600 topographic map with resolution 2m/pixel on an Intel Xeon 1.8 GHZ Dual Processors server with 1 GB memory. Other sources need less than 20 seconds on images smaller than 500 x 400.

4. RELATED WORK

There is a variety of research on extracting information (i.e., road intersection extraction, building recognition, contour line extraction) from raster maps [8, 9, 11, 12] and satellite imagery [1]. The problem of extracting information from satellite imagery is more difficult than for raster maps, thus the techniques (e.g., road tracking and grouping [14]) used are more computationally intensive. Since our technique deals only with raster maps, we focus our comparison on related work in extracting information from raster maps.

The approaches in [8, 9, 11, 12] to exploit the input raster maps rely on a variety of prior knowledge. The main difference between our approach and the previous work is that we assume a more general situation where we do not have any prior knowledge about how to separate the road layers from other layers in the input raster maps, such as the color of the road lines, legend information, etc., and the road layers on raster maps have not been extracted manually before road intersection extraction.

Salvatore and Guitton [11] use color classification to separate contour lines from other objects on the topographic maps and apply image processing algorithms with global topology information to reconstruct the broken lines, which requires prior knowledge and experiments to generate a proper set of color thresholds to separate the contour lines from other objects. With our approach, the system does not have prior knowledge of the road line color in the input raster map. We use an automatically generated threshold to separate the foreground pixels, which include road lines, and utilize the text/graphics separation algorithm with morphological operators to extract road lines from foreground pixels. Moreover, for the road-layer extraction step, in the previous work the goal is to ensure that the resulting contour lines have a continuity close to the original, which makes the problem hard to solve and the time complexity higher compared to the use of morphological operators in our approach. We focus on the road lines close to each intersection point, and ignore the accuracy of the entire road layer to save computation time. The drawback of morphological operators is that they do not guarantee that the road lines have the same continuity as before the text/graphics separation. This does not cause a problem in our approach as we are only interested in segments around each intersection point and the broken lines usually occur in the middle of road lines and not around the intersection points.

Habib et al.[8] utilize several image processing methods to automatically extract primitives on raster maps. They detect the corner points (i.e., salient points in our paper) of the foreground objects by determining the magnitude and orientation array using an edge detector and an interest operator. However, they require the input raster maps contain no characters or labels, and there are major drawbacks using this method in our automatic approach when there are other layers in the input raster maps. First, the edge detector is sensitive to noise, which makes it difficult to determine the threshold automatically in raster maps with many objects and more than one color in the background. Second, the edge detector usually makes the resulting characters fatter than the original ones. Fat characters touch more road lines and they are harder to remove.

Samet et al.[12] use the legend layer in a learning process to identify labels on the raster maps. Meyers et al.[9] use a verification based approach to extract data on raster maps, which require map specifications and legends. These approaches all need prior knowledge of the input raster maps, such as the color of objects that need to be extracted or the legend information.

5. CONCLUSION AND FUTURE WORK

The main contribution of this paper is to provide a framework to automatically and efficiently extract road intersections from arbitrary raster maps by combining several well-studied image processing and graphic recognition algorithms. Our approach achieves 92% precision and 77% recall when automatically extracting road intersection points with no prior information on the input raster maps (resolution higher than 7m/pixel). The resulting point sets provide accurate features for identifying the input raster maps. For example, a conflation system [4] can utilize this technique to discover geospatial data (such as other raster maps, imagery, and vector data) with the same coverage as the given raster map. In addition, we apply our technique on randomly returned maps from image search engines and successfully extract the road intersection points for conflation systems to identify the geocoordinate [6].

We have made two general assumptions about the input raster maps. Firstly, the background pixels must be separable using the difference of luminosity level from the foreground pixels, which contain road, building, symbol, and character layers as well as any other notations. This means that the background pixels must have the dominant color in the raster maps. On certain raster maps that contain numerous objects and the number of foreground pixels is larger than that of the background pixels, the information layers overlap each other, which makes the automatic processing nearly impossible. Even if we can remove the background pixels on these raster maps, removing noisy objects touching road lines will break the road layer into small pieces which is hard to reconnect. Secondly, although our approach works with no prior knowledge of the map scales, low-resolution raster maps may lead to low precision and recall.

We intend to extend our work in several ways. Firstly, we want to handle other types of raster maps, such as low-quality scanned maps with more than one luminosity level on their background pixels. In this case, we need to improve the automatic segmentation component with histogram analysis to generate the threshold in order to separate the foreground pixels from the background pixels. Secondly, we plan to add different morphological operators to raise the recall of our approach. There are other morphological operators which have similar effect as the ones in our current approach. For example, the skeletonizing operator produces one pixel width results similar to those of the thinning operator but the resulting shapes of the road layers will be different. The differences will impact the precision and recall of the final result, and users can choose to raise either precision or recall by using other morphological operators.

6. ACKNOWLEDGMENTS

This research is based upon work supported in part by the National Science Foundation under Award No. IIS-0324955, and in part by the Air Force Office of Scientific Research under grant number FA9550-04-1-0105. 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.

7. REFERENCES

[1] Marie-Flavie Aculair-Fortier, Djemel Ziou, Costas Armenakis, and Shengrui Wang. Survey of work on road extraction in aerial and satellite images. Technical report, Universite de Sherbrooke, 2000.

[2] J. Patrick Bixler. Tracking text in mixed-mode documents. In *ACM Conference on Document processing systems*, 2000.

[3] Ruini Cao and Chew Lim Tan. Text/graphics separation in maps. In *the Fourth International Workshop on Graphics Recognition Algorithms and Applications(GREC 2001)*, 2001.

[4] Ching-Chien Chen. *Automatically and Accurately Conflating Road Vector Data, Street Maps and Orthoimagery*. PhD thesis, University of Southern California, 2005.

[5] Ching-Chien Chen, Craig A. Knoblock, Cyrus Shahabi, Yao-Yi Chiang, and Snehal Thakkar. Automatically and accurately conflating orthoimagery and street maps. In *The 12th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS'04)*, 2004.

[6] Sneha Desai, Craig A. Knoblock, Yao-Yi Chiang, Kandarp Desai, and Ching-Chien Chen. Automatically identifying and georeferencing street maps on the web. In *The 2nd International Workshop on Geographic Information Retrieval (GIR'05)*, 2005.

[7] Lloyd Alan Fletcher and Rangachar Kasturi. A robust algorithm for text string separation from mixed text/graphics images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):910–918, 1988.

[8] Ayman F. Habib and Robert E. Uebbing. Automatic extraction of primitives for conflation of raster maps. Technical report, The Center for Mapping, The Ohio State University, 1999.

[9] Gregory K. Myers, Prasanna G. Mulgaonkar, Chien-Huei Chen, Jeff L. DeCurtins, and Edward Chen. Verification-based approach for automated text and feature extraction from raster-scanned maps. In *Lecture Notes in Computer Science*, volume 1072, pages 190–203. Springer, 1996.

[10] William K. Pratt. *Digital Image Processing: PIKS Inside*. Wiley-Interscience, third edition edition, 2001.

[11] Spinello Salvatore and Pascal Guitton. Contour line recognition from scanned topographic maps. Technical report, University of Erlangen, 2001.

[12] Hanan Samet and Aya Soffer. A legend-driven geographic symbol recognition system. In *12th International Conference on Pattern Recognition*, volume 2, pages 350–355, 1994.

[13] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[14] Carsten Steger, Helmut Mayer, and Bernd Radig. The role of grouping for road extraction. *Automatic Extraction of Man-Made Objects from Aerial and Space Images(II)*, 245-256:1931–1952, 1997.

[15] Yuan Yan Tang, Seong-Whan Lee, and Ching Y. Suen. Automatic document processing: A survey. *Pattern Recognition*, 29(12):1931–1952, 1996.

[16] Aurelio Velázquez and Serguei Levachkine. Text/graphics separation and recognition in raster-scanned color cartographic maps. In *the Fifth International Workshop on Graphics Recognition Algorithms and Applications(GREC2001)*, 2003.

[17] G.W. Zack, W.E. Rogers, and S.A. Latt. Automatic measurement of sister chromatid exchange frequency. *Journal of Histochemistry and Cytochemistry*, 25(7):741–753, 1977.