Extracting Layers and Recognizing Features for Automatic Map Understanding

Yao-Yi Chiang
Outline

- Introduction/ Problem Motivation
- Map Processing Overview
- Map Decomposition
- Feature Recognition
- Discussion
Raster Maps

- The raster map is one important source of geospatial data:
  - Contain information that is difficult to find elsewhere
  - Contain the most complete set of data
Exploit information in raster maps

- Align raster map with other geospatial data
- Label other geospatial data with map features
Difficulties

- Maps are complex
- Limited access to the meta-data
  - Scanned Thomas Guide map
  - USGS topographic map
  - Rand McNally map
Outline

- Introduction/ Problem Motivation
- Map Processing Overview
- Map Decomposition
- Feature Recognition
- Discussion
Map Processing Overview

- **Earlier work focuses on hardware**
- **Recent work**
  - Map processing system using map legends and map spec.
  - **Work on a specific type of maps**
    - Work on maps with layers can be extracted using color or grayscale thresholding
    - Work on digital maps only
    - Work on maps with only the target layers
  - Supervised map processing with user labeling
Map processing in the earlier years

- From paper maps to vector data (Leberl and Olson, 82; Suzuki and Yamada, 90):
  - Highly labor-intensive
  - Focus on hardware and the techniques to help human operators to speed up the processes

Source: Leberl, 82
Map processing system using map legends

MAGELLAN: Samet and Soffer, 98
Map processing system using prior knowledge

- Verification-based computer vision approach (Myers et al., 96)
- “A verification-based approach uses contextual knowledge and constraints to formulate and then verify interpretation hypotheses.”

Source: Myers et al., 96

Figure 1. Automated Map Data Extraction Architecture
Extract layers using color thresholding

- Extraction and recognition of geographical features from paper maps (Dhar and Chanda, 06)

**Green layer**

**Red layer**
- Integrated text and line-art extraction from a topographic map (Li et al., 00)
- Use connected component analysis to separate lines and characters and focus on local area to rebuild lines

Source: Li et al., 00
Work on digital maps only

- Automatic extraction of road networks from map images (Itonaga et al., 03)
- Label initial road areas using geometrical properties
- Iteratively update the probability of area labels until it converges
Work on maps with roads only

- **Automatic extraction of primitives for conflation of raster maps.** (Habib et al. 1999)
  - Require the input raster maps have only road layer
  - Automatically extract primitives on raster maps
  - Group corner points and extract the centroid as the road intersections

Source: Habib et al., 99
Supervised map processing with user labeling

- Label all combination of background colors and line colors
- Contour line extraction (Khotanzad and Zink, 03, Chen et al. 06)

<table>
<thead>
<tr>
<th>Combination Color Key</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black on white</td>
<td><img src="image1" alt="Samples" /></td>
</tr>
<tr>
<td>Black on green</td>
<td><img src="image2" alt="Samples" /></td>
</tr>
<tr>
<td>Brown on white</td>
<td><img src="image3" alt="Samples" /></td>
</tr>
<tr>
<td>Brown on green</td>
<td><img src="image4" alt="Samples" /></td>
</tr>
<tr>
<td>Blue on white</td>
<td><img src="image5" alt="Samples" /></td>
</tr>
<tr>
<td>Blue on green</td>
<td><img src="image6" alt="Samples" /></td>
</tr>
<tr>
<td>Purple on white</td>
<td><img src="image7" alt="Samples" /></td>
</tr>
<tr>
<td>Purple on green</td>
<td><img src="image8" alt="Samples" /></td>
</tr>
</tbody>
</table>

Source: Khotanzad and Zink, 03
Automatic Map Understanding

Raster Map

Map Decomposition

Road Layer (raster)

Text Layer (raster)

Text Recognition

Road Intersection Extraction

Road Intersection Templates

Road Vectorization

Transformation Matrix
(Chen et al. 08)
• Scaling
• Rotation
• Translation

Georeferenced Sources

Road (vector)

Lat / Long

Map Context
1. **Automatic** Map Decomposition

2. **Supervised** Map Decomposition
   2.1 Automatic Map Classification
      - Reuse training results
      - Reduce human efforts

3. Automatic Feature Recognition
   3.1 Road Intersection Extraction
   3.2 Road Vectorization
   3.3 Text Recognition
Outline

- Introduction/ Problem Motivation
- Map Processing Overview
- Map Decomposition
  - Automatic Approach
    - Extraction of Road and Text Layers
  - Supervised Approach
- Feature Recognition
- Discussion
Background Removal

- Use Triangle method (Zack, 1977) to locate clusters in the grayscale histogram
- Remove the dominate cluster

Remove dominate cluster (background pixels)
Text/Graphics Separation

- Separate linear structures from text (Cao and Tan, 02)

Add up the removed objects
Identify Road-Layer Format

- Double-line road layer and single-line road layer
Parallel-Pattern Tracing

- Assuming we know the road width is 3 pixels, if the yellow pixel is on a double-line layer, we can find:
  - At least 1 pixel on the original road line
  - At least 1 corresponding pixel on the other road line

The parallel pattern!

Pixilated view of a segment of double-line streets
Black cells: Road pixels
White cells: Backgrounds

Corresponding pixel on the second line
Construct the first line
Identify Road Format and Road Width

- Apply parallel-pattern tracing (PPT) iteratively on different sizes of road width
- The actual road width
  - Has the maximum percentage of parallel pattern pixels
  - The percentage is larger than a threshold

Examples are in the next slide
Identify road width and road-layer format

![Graph showing the relationship between road width and parallel pixel ratio for different map services. The x-axis represents the width in pixels, and the y-axis represents the parallel pixel ratio (identified parallel-line pixels / foreground pixels). The graph compares ESRI Map High Resolution, MapQuest Map High Resolution, Yahoo Map High Resolution, and USGC Topographic Map.]

- Double-line maps
- Single-line maps

[X-axis] Width in pixels
[Y-axis] Parallel Pixel Ratio (Identified parallel-line pixels / Foreground pixels)
Remove non-parallel lines

Apply PPT using the detected road with to remove non-parallel lines
General Dilation Operator

- Reconnect the broken road layer

For every foreground pixel, fill up its eight neighborhood pixels.

After 2 iterations

2nd iteration
General Erosion Operator

- Thinner road lines and maintain the orientation

For every foreground pixel, erase itself if any of its neighboring pixel is a background pixel.

After 2 iterations

2nd iteration
Thinning Operator

- Produce one-pixel width road lines

Reduce all lines to single-pixel thickness.
Outline

- Introduction/ Problem Motivation
- Map Processing Overview
- Map Decomposition
  - Automatic Approach
  - Supervised Approach
    - Extraction of Road Layer
- Feature Recognition
- Discussion
Supervised Map Decomposition

- What if we cannot automatically remove the background from raster maps?
- Raster maps usually contain noise from scanning and compression process
Difficulties

- Raster maps contain numerous colors
  - Two hundred and eighty five thousand seven hundred and thirty five colors

285,735 colors
Color Segmentations

- **The Mean-shift algorithm**
  - Consider distance in the RGB color space and in the image space
  - Preserve object edges
  - From 285,735 to 155,299 colors

- **The K-means algorithm**
  - Limit the number of colors to K
  - From 155,299 to 10 colors (K=10)
User Labeling

To extract the road layer, the user needs to provide a user label for each road color (at most K colors)

User label should be (approximately) centered at a road intersection or at the center of a road line.
Label Decomposition

- Decompose each user label into color images so that every color image contains only one color

(Background is shown in black)
Hough Transformation

- The simplest case of Hough transform is the linear transform for detecting straight lines
- Line function in Hough Transformation
  \[ y = \left( -\frac{\cos \theta}{\sin \theta} \right) x + \left( \frac{r}{\sin \theta} \right) \]

- \( r \) represents the distance between the line and the origin
- \( \theta \) is the angle of the vector from the origin to this closest point
- An infinite number of lines can pass through a single point \((x_0,y_0)\) of the plane
- All the lines that go through \((x_0,y_0)\) obey the following equation:
  \[ r(\theta) = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta \]

Hough Transformation for Line Detection
Hough-Line Approach to Identify Road Color

- Detect Hough Lines
  - Red Hough lines are within 5 pixels to the image center
- The center of the image is a road intersection or the center of a road line
- Identify road colors using
  - the average distance between the Hough lines to the image center

(Background is shown in black)
Generate Road Template

- Generate a road template using the images of identified road colors

![Image of road generation process]

(background is shown in black)

(road pixels are shown in red, background is shown in black)
Template Matching Approach to Identify Road Color

- More than 50% of the red pixels in the test image have found a match – identified as an image of road color.
Color Filter

- Identify a set of road colors from each user label
- Add the identified road colors to generate a color filter
- Use the color filter to extract the road layer
Extract one-pixel width road layer

- Apply morphological operations to remove solid areas and reconnect lines
Experimental Results

- We successfully extracted the road layers from 100 maps.
- The average number of user labels per map was under 4.
- The average computation time per map was below 5 seconds.

<table>
<thead>
<tr>
<th>Map Set</th>
<th>Map Count</th>
<th>Avg. User Labels</th>
<th>Avg. Computation Time (s.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Scanned Iraq maps</td>
<td>12</td>
<td>2.7</td>
<td>2.7</td>
</tr>
<tr>
<td>2. Scanned Iraq maps</td>
<td>18</td>
<td>2.1</td>
<td>3.3</td>
</tr>
<tr>
<td>3. Scanned Iraq maps</td>
<td>30</td>
<td>2.6</td>
<td>4.2</td>
</tr>
<tr>
<td>4. USGS topo. maps</td>
<td>19</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>5. USGS topo. maps</td>
<td>5</td>
<td>1.6</td>
<td>1.2</td>
</tr>
<tr>
<td>6. USGS topo. maps</td>
<td>6</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>7. Rand McNally maps</td>
<td>10</td>
<td>3.9</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Map Classification

- Can we automatically select a color filter for a new map?

New map

Map Repository

Color Filters
Content-based Image Retrieval

Comparing raster map content (i.e., image content) using:

- **Color Histogram**
- **Color Coherence Vectors**
- **Edge-Intensity Histograms**

Use Color Information Only

Use Spatial Information of the Color Pixels Only
Image Features

- **Color Histogram**
  - Record the number of pixels per color

- **Color Coherence Vectors**
  - Record the number of pixels per color based on the sizes of color regions

- **Edge-intensity histograms**
  - Record the spatial relationships between different intensities
Experimental Setup

- Can we identify maps that share the same intensity threshold?

<table>
<thead>
<tr>
<th>Map Source</th>
<th>Map Type</th>
<th>Map Counts</th>
<th>Intensity Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Maps</td>
<td>Digital</td>
<td>5</td>
<td>0–230</td>
</tr>
<tr>
<td>Live Maps</td>
<td>Digital</td>
<td>5</td>
<td>0–225</td>
</tr>
<tr>
<td>Yahoo Maps</td>
<td>Digital</td>
<td>5</td>
<td>0–200</td>
</tr>
<tr>
<td>MapQuest Maps</td>
<td>Digital</td>
<td>5</td>
<td>0–220</td>
</tr>
<tr>
<td>USGS topographic maps</td>
<td>Scanned</td>
<td>5</td>
<td>0–36</td>
</tr>
<tr>
<td>USGS topographic maps</td>
<td>Scanned</td>
<td>5</td>
<td>0–184</td>
</tr>
<tr>
<td>Rand McNally</td>
<td>Digital</td>
<td>5</td>
<td>0–190</td>
</tr>
<tr>
<td>Map24</td>
<td>Digital</td>
<td>5</td>
<td>0–215</td>
</tr>
<tr>
<td>TIGER/Line</td>
<td>Digital</td>
<td>5</td>
<td>0–110</td>
</tr>
<tr>
<td>OpenStreetMap</td>
<td>Digital</td>
<td>5</td>
<td>0–238</td>
</tr>
<tr>
<td>Streetmap.co.uk</td>
<td>Digital</td>
<td>5</td>
<td>0–175</td>
</tr>
<tr>
<td>ViaMichelin</td>
<td>Digital</td>
<td>5</td>
<td>0–234</td>
</tr>
</tbody>
</table>

- Insert the test maps into a repository with 5,502 images
- Take one test map out and use it to query the repository (CBIR)
- Check if the first returned map is in the same class as the query map

The grayscale thresholds to extract foreground pixels
Experimental Results

- We successfully classified over 80% of the maps using three of the tested features.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color Histogram</td>
<td>83.33%</td>
</tr>
<tr>
<td>Color-Coherence Vectors</td>
<td>80%</td>
</tr>
<tr>
<td>Edge-Intensity Histograms</td>
<td>93.33%</td>
</tr>
</tbody>
</table>

The two maps share the same intensity threshold (175) to extract their foreground pixels.
Outline

- Introduction/ Problem Motivation
- Map Processing Overview
- Map Decomposition
- Feature Recognition
  - Road-Intersection Position Detection
  - Road-Intersection Template Extraction
  - Road Vectorization
  - Text Recognition
- Discussion
Road-Intersection Position Detection

- Corner detector (OpenCV)
  - Find intersection candidates
  - Compute the connectivity to determine real intersections

Corner Detector

Connectivity < 3, discard

Connectivity ≥ 3

Road Intersection!!
Road-Intersection Position Detection Evaluation Metrics

\[
\text{Precision} = \frac{\text{Num. of Correctly Extracted Road Intersections}}{\text{Num. Extracted of Road Intersections}}
\]

\[
\text{Recall} = \frac{\text{Num. of Correctly Extracted Road Intersections}}{\text{Num. of Road Intersections}}
\]
Experimental Results

- We tested 77 maps from 12 sources to detect the road intersections
  - The **average precision is 95% and the recall is 75%** for the road intersection detection

<table>
<thead>
<tr>
<th>Map Source (Number of Test Maps)</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESRI Maps (8)</td>
<td>93%</td>
<td>71%</td>
<td>81%</td>
</tr>
<tr>
<td>MapQuest Maps (9)</td>
<td>98%</td>
<td>66%</td>
<td>79%</td>
</tr>
<tr>
<td>TIGER/Line Maps (8)</td>
<td>97%</td>
<td>84%</td>
<td>90%</td>
</tr>
<tr>
<td>Yahoo Maps (10)</td>
<td>95%</td>
<td>76%</td>
<td>84%</td>
</tr>
<tr>
<td>A9 Maps (5)</td>
<td>100%</td>
<td>93%</td>
<td>97%</td>
</tr>
<tr>
<td>MSN Maps (5)</td>
<td>97%</td>
<td>88%</td>
<td>92%</td>
</tr>
<tr>
<td>Google Maps (5)</td>
<td>98%</td>
<td>86%</td>
<td>91%</td>
</tr>
<tr>
<td>Map24 Maps (5)</td>
<td>100%</td>
<td>82%</td>
<td>90%</td>
</tr>
<tr>
<td>ViaMichelin Maps (5)</td>
<td>100%</td>
<td>98%</td>
<td>99%</td>
</tr>
<tr>
<td>Multimap Maps (5)</td>
<td>98%</td>
<td>85%</td>
<td>91%</td>
</tr>
<tr>
<td>USGS Topographic Maps (10)</td>
<td>82%</td>
<td>60%</td>
<td>69%</td>
</tr>
<tr>
<td>Thomas Brothers Maps (2)</td>
<td>98%</td>
<td>65%</td>
<td>79%</td>
</tr>
</tbody>
</table>
Experimental Results (with respect to the positional displacement)

Distance in pixels to the ground truth
Road-Intersection Template Extraction

- Road-intersection template
  - road intersection position
  - road connectivity
  - road orientation
- Road lines are distorted by the thinning operator
- The extracted road-intersection templates will not be accurate
Avoid Distortion

Use the road width to determine the blob size for covering the distorted lines.

Intersection Positions

The thinned lines

Intersect the images
Trace road line candidates from contact points

- Trace only a certain amount of line pixels
  - Road lines are straight within a small distance (e.g., 5 pixels)
- Fit a line function ($Y = aX + b$) to the traced pixels using Least-Squares Fitting algorithm
Update road-intersection templates – Case I

- Keep every road line candidate
- Use the intersection of the line candidates to update the template
Update road-intersection templates – Case II

- Update road intersection templates
- Keep every road line candidate
- Use the centroid of the intersections of the line candidates to update the template
Update road-intersection templates – Case III

- Remove outliers and use the centroid of remaining intersections
Accurate Road-Intersection Templates

Avoid distortion

With distortion
Road-Intersection Template Quality Metrics

- The blue lines are the extracted road-intersection templates and the red lines are the ground truth.
Road-Intersection Template Quality Results

- Test 10 maps
  - Extracted 139 road-intersection templates with 438 lines

- The average positional offset:
  - 0.4 pixels

- The average orientation offset:
  - 0.24 degrees

- The connectivity offset:
  -Missed 13 lines from a total of 451 lines (3%)
Outline

- Introduction/ Problem Motivation
- Map Processing Overview
- Map Decomposition
- Feature Recognition
  - Road-Intersection Positions Detection
  - Road-Intersection Template Extraction
  - Road Vectorization
- Text Recognition
- Discussion
Road Vectorization

- Start from the extracted road intersections to connect the salient points and produce the road vector.
Discussion

- We can exploit the information in raster maps by decomposing the maps into different feature layers, recognizing features from the raster layers, and aligning the features to other geospatial data.
- We extract the road-intersection templates, the road vector, and text labels.
- The extracted road-intersection templates and road vector can be used to align the raster map with imagery.
Questions?

Thank You

- If you are interested in any of the referred papers in the slides, please send me an email, and I will reply you with a pointer to the paper