A General Approach to Discovering, Registering, and Extracting Features from Raster Maps

Craig Knoblock
University of Southern California
&
Geosemble Technologies

Joint work with Ching-Chien Chen, Yao-Yi Chiang, Aman Goel, Matthew Michelson, and Cyrus Shahabi
Introduction

- Raster maps are a rich source of geospatial data:
  - Easily accessible
  - Many different types of information
  - Often contains information that cannot be found elsewhere

Travel map of Tehran, Iran

USGS topographic map of St. Louis, MO
Challenges

• Maps have lots of useful information, but...
  — They have overlapping features
  — There is limited access to the meta-data
  — Often only available in raster format

• How do we find, register, and extract and recognize the features in a raster map
Outline

- Map Discovery
- Automatic Extraction of Features
- Feature Extraction from Noisy Maps
- Automatic Registration of Maps
- Next Steps
- Related Work and Discussion
• Map Discovery
• Automatic Extraction of Features
• Feature Extraction from Noisy Maps
• Automatic Registration of Maps
• Next Steps
• Related Work and Discussion
Map Discovery

- Collect candidate maps from the Web
  - Standalone maps
    - Found using an image search engine
  - Maps embedded in PDF documents
    - Found using a general search engine and then extracting the images

- Classify the images
  - Extract features from the images
  - Identify similar images using Content Based Image Retrieval (CBIR)
  - Classify the image using k-Nearest Neighbor
Identifying Maps

Our approach:

- Extract features
- Find similar images
- Classify image
• **Water-filling features**
  

  — Works well on images with strong edges

  o Works on standard Canny edge maps of original images
    - Color invariant
- Features computed for each segment

  Fork Count: 0
  Filling Time: 45
  Water Amount: 45

  Fork Count: 6
  Filling Time: 57
  Water Amount: 68

- Normalized histogram - size invariant

- 3 features x 8 buckets = 24 element feature vector
Content-Based Image Retrieval (CBIR)

Map repository

Non-map repository

CBIR* (find 5 most similar images)

Query image feature vector

- Map12
- Map75
- Map36
- Non-map23
- Non-map139

- Built on top of Lire system

* In our experiment we used 9 similar images
k - Nearest neighbor classification

Map12  Map75  Map36  Non-map23  Non-map139

Majority Maps?  yes  Label image as a map
Results are average over 10 runs

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
<th>F1-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.39%</td>
<td>71.20%</td>
<td>74.17%</td>
</tr>
</tbody>
</table>

8,000 images (4,000 maps/4,000 nonmaps)

4,000 images (2,000 maps/2,000 nonmaps)

Repository

Test set
Outline

- Map Discovery
- **Automatic Extraction of Features**
- Feature Extraction from Noisy Maps
- Automatic Registration of Maps
- Next Steps
- Related Work and Discussion
• Use the Triangle method (Zack, 1977) to locate clusters in the grayscale histogram
• Remove the background clusters
• Separate linear structures from text (Cao and Tan, 02)

Add up the removed objects
Road Format and Road Width Detection

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

Apply PPT using the detected road with to remove non-parallel lines
Road Topology Extraction

- Use morphological operations to reconnect broken lines and generate one-pixel width roads

Morphological Operations: Use the detected road format and road width to determine the number of iterations

- Dilation
- Erosion
- Thinning
Extracted Road and Text Layers
Outline

- Map Discovery
- Automatic Extraction of Features
- **Feature Extraction from Noisy Maps**
- Automatic Registration of Maps
- Next Steps
- Related Work and 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
  - Manually examining each color for extracting features is laborious

285,735 colors
• **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)
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.
Decompose each user label into color images so that every color image contains only one color.

(background is shown in black)
Hough-Line Approach to Identify Road Color

- Detect Hough lines
- The center of the user label is the center of a road line
  - The Hough lines that are far away from the image center are NOT constructed by road pixels
- Identify road colors using
  - The average distance between the Hough lines to the image center

Red Hough lines are within 5 pixels to the image center
• Generate an initial road template using the images of identified road colors from the Hough line approach.

(road pixels are shown in red, background is shown in black)
Road Topology Extraction using Identified Road Colors

- Identify a set of road colors from each user label
- Use the identified road colors to extract road pixels
- Apply morphological operations to remove solid areas and reconnect lines
Outline

• Map Discovery
• Automatic Extraction of Features
• Feature Extraction from Noisy Maps
• **Automatic Registration of Maps**
• Next Steps
• Related Work and Discussion
Automatic Map Registration

- Exploit the pattern of intersections found on a map and compare to a road vector dataset
• 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
Road-Intersection Position Detection

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

Connectivity<3, discard

Connectivity>=3

Corner Detector

Road Intersection!!
Distortion Correction

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

Intersect the images

Intersection Positions

The thinned lines
Accurate Road-Intersection Templates

With distortion

Avoid distortion
Next Steps: Road Vectorization

- Start from the extracted road intersections to connect the salient points and produce the road vector
Next Steps: Text Recognition

- Generalize OCR techniques to apply to maps
  - Identify individual characters regardless of orientation
  - Exploit background knowledge to improve accuracy
Outline

- Map Discovery
- Automatic Extraction of Features
- Feature Extraction from Noisy Maps
- Automatic Registration of Maps
- Next Steps
- Related Work and Discussion
Related Work

- **Map Feature Extraction Using Map Specification** (Samet and Soffer, 94, 96; Myers et al., 96)
  - Require huge amount of prior information and training
- **Text/Graphics Separation and Text Recognition** (Bixler, 00; Li et al., 00; Cao and Tan, 02; Vela, 03; Pouderoux, 07)
  - Require fixed pre-processing steps, e.g., binarization with fixed threshold
- **Supervised Graphics Extraction** (Khotanzad and Zink, 03; Salvatore and Guitton, 04; Chen et al., 06)
  - Laborious training tasks, e.g., labeling all combination of line and background pixels
- **Road Extraction and Vectorization** (Bin, 98; Habib et al., 99; Itonaga et al., 03)
  - Require lots of parameter tunings, e.g., road width
- **Map, Vectors, and Imagery Conflation** (Chen et al., 06; Chen et al., 08; Wu et al., 07)
  - Exploit for determining feature locations
Discussion

• Presented a general approach to discovering, registering, extracting features from maps

• Contributions
  — Ability to identify maps
  — Ability to extract road and text layers
  — Automatic recognition of road intersection
  — Algorithms to automatically determine the geocoordinates

• Applications
  — Annotating imagery
  — Creating and updating maps
  — Constructing gazetteers