Automatic Extraction of Road Intersections from Raster Maps

Yao-Yi Chiang, Craig A. Knoblock
and Ching-Chien Chen

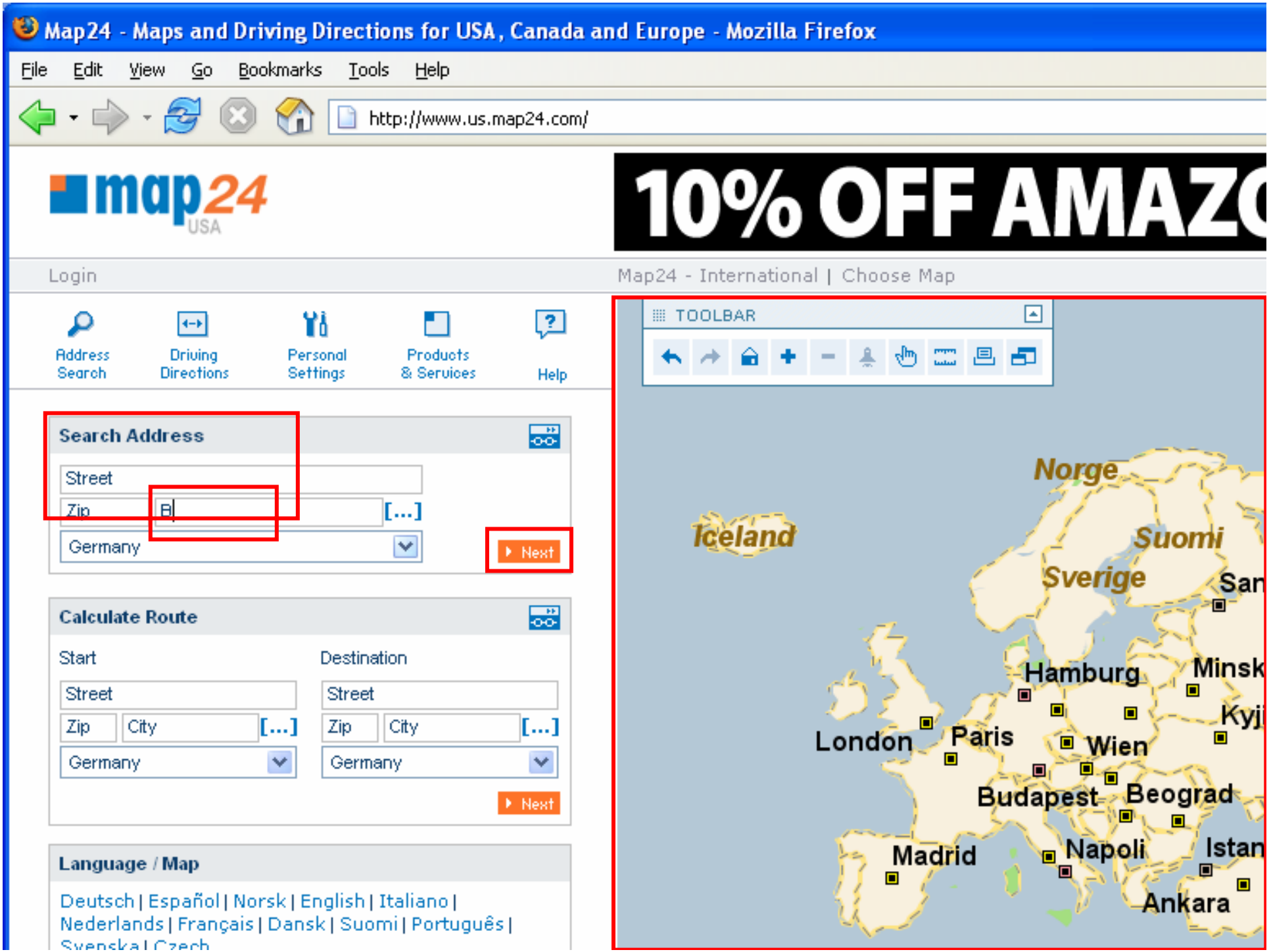
University of Southern California
Department of Computer Science and Information Sciences Institute

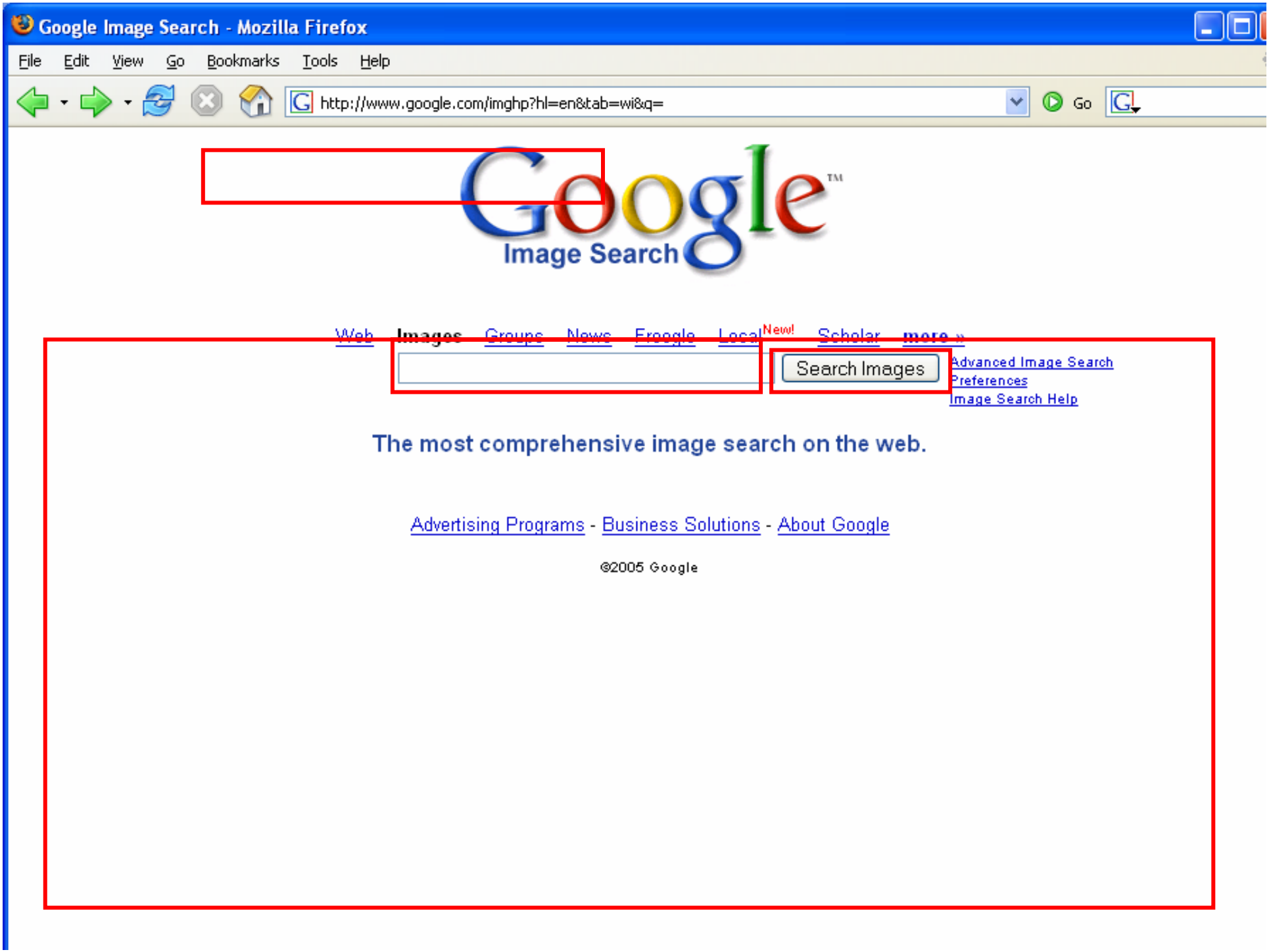
Outline

- Introduction and Motivation
- Approach and Algorithm
- Experimental Results
- Related Work
- Conclusion and Future Work

Introduction and Motivation

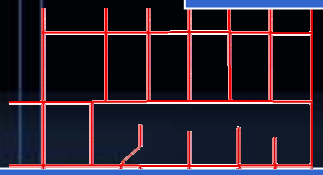
- Numerous raster maps are on the Internet
  - Online map provider:
    - Google Map, Yahoo Map, USGS Topographic Map, Map24
  - Image Search Engine:
    - Google Image, MSN Image
- The georeferencing information of them are often unknown

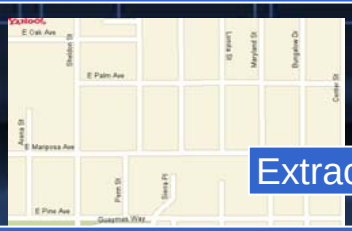

Map24 - Maps and Driving Directions for USA, Canada and Europe - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.us.map24.com/
map24 USA
10% OFF AMAZ
Login
Map24 - International | Choose Map
Address Search
Driving Directions
Personal Settings
Products & Services
Help
Search Address
Street
Zip
[...]
Germany
Next
Calculate Route
Start
Destination
Street
Zip
City
[...]
Germany
Next
Language / Map
Deutsch | Español | Norsk | English | Italiano |
Nederlands | Français | Dansk | Suomi | Português |
TOOLBAR
Norge
Iceland
Suomi
Sverige
Hamburg
Minsk
London
Paris
Wien
Budapest
Beograd
Madrid
Napoli
Ankara

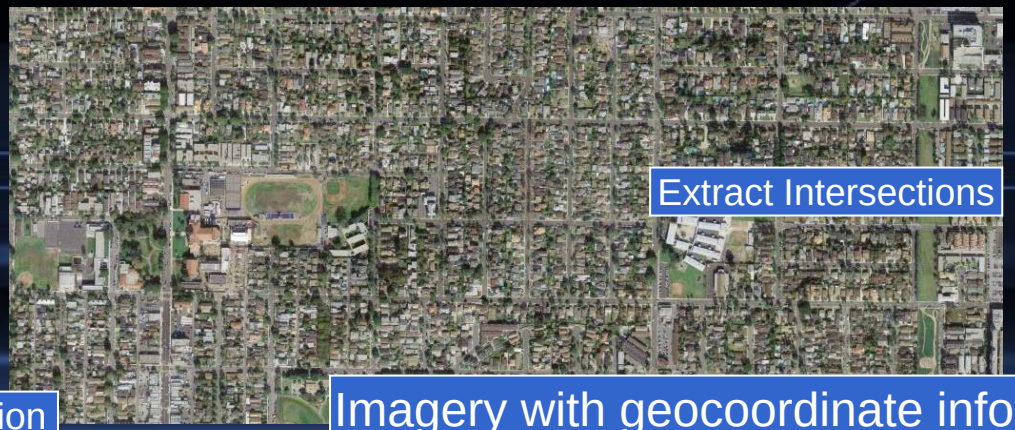

Google Image Search - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.google.com/imghp?hl=en&tab=wi&q=
Go
Google™
Image Search
Web Images Groups News Froogle Local New! Scholar more »
Search Images
Advanced Image Search
Preferences
Image Search Help
The most comprehensive image search on the web.
Advertising Programs - Business Solutions - About Google
©2005 Google

Introduction and Motivation

- In our previous work: Automatically and Accurately Conflating Orthoimagery and Street Maps (Chen et al.)
  - We utilize the layout of the road intersections within a local area to
    - Integrate imagery, raster maps and vector data
      - Align street lines from each source
      - Georeference raster map

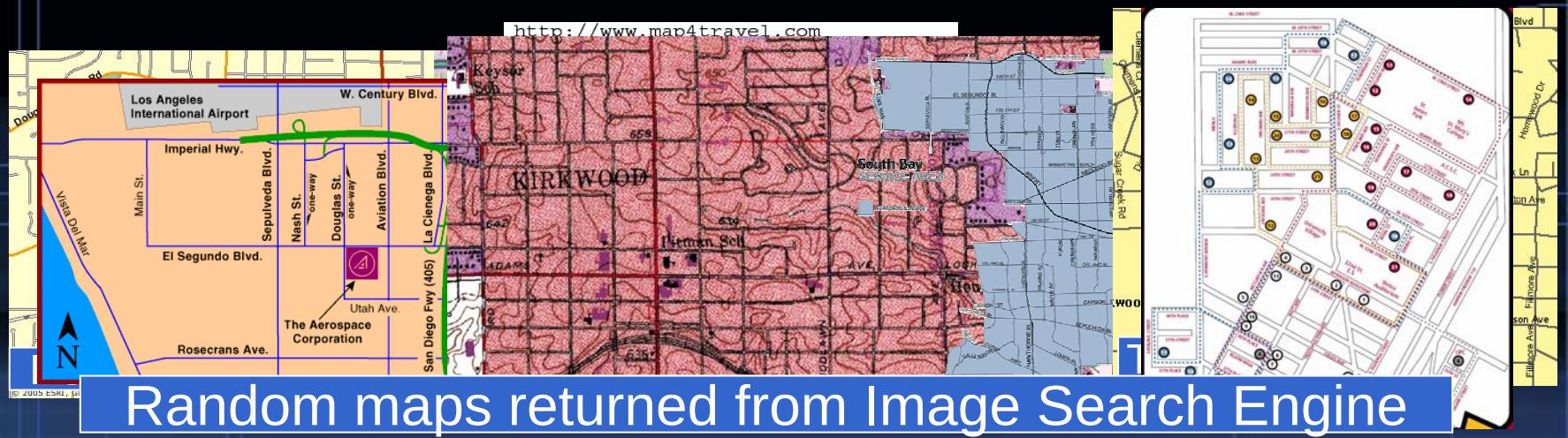




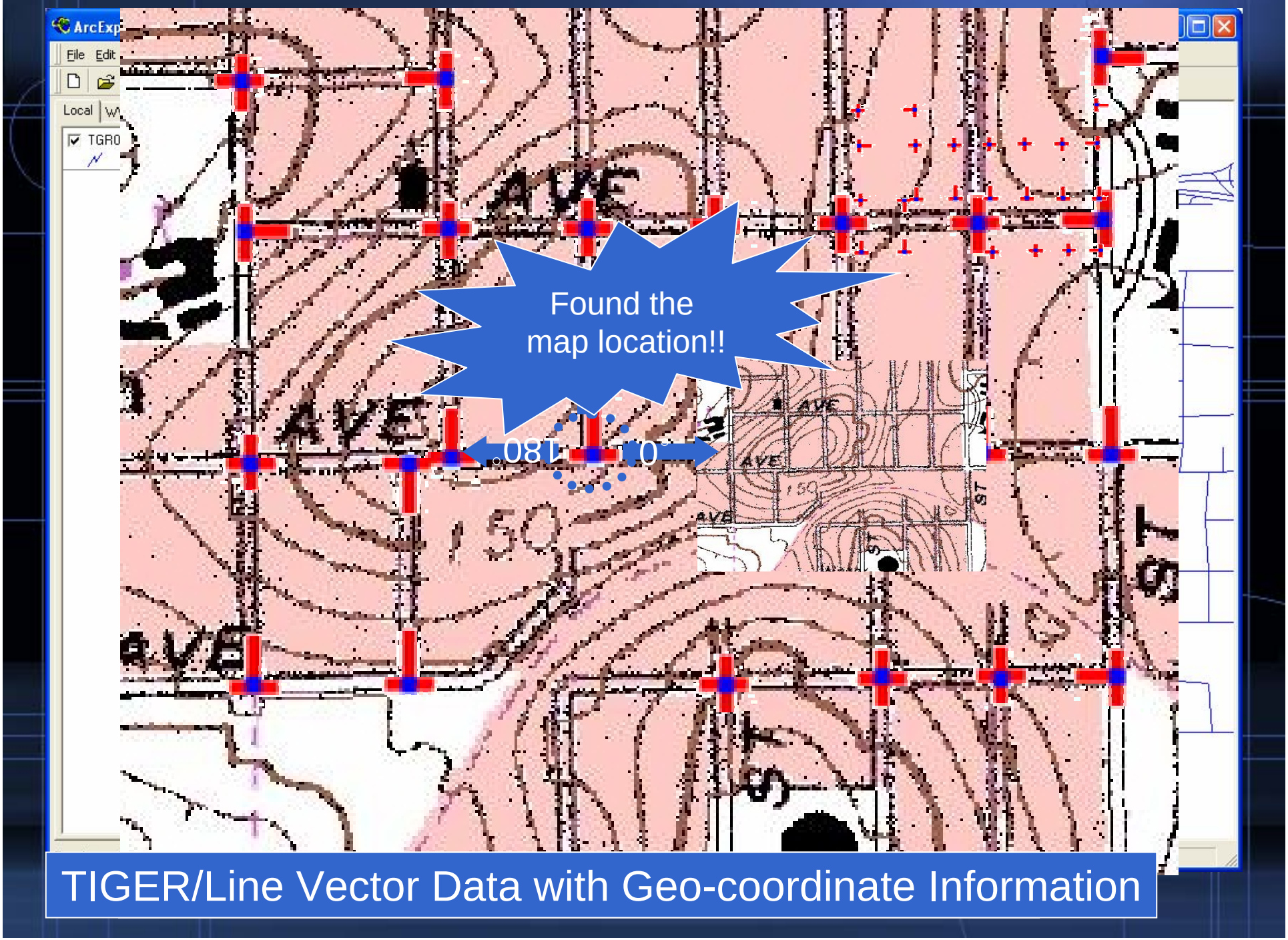


Introduction and Motivation

- The correct road intersection pattern is important!
- More information about the road intersection is important!
- In this work:
  - The average precision of intersection extraction is improved from 76% to 92%.
  - Extract road information around each intersection point
  - Handle more types of map



Random maps returned from Image Search Engine



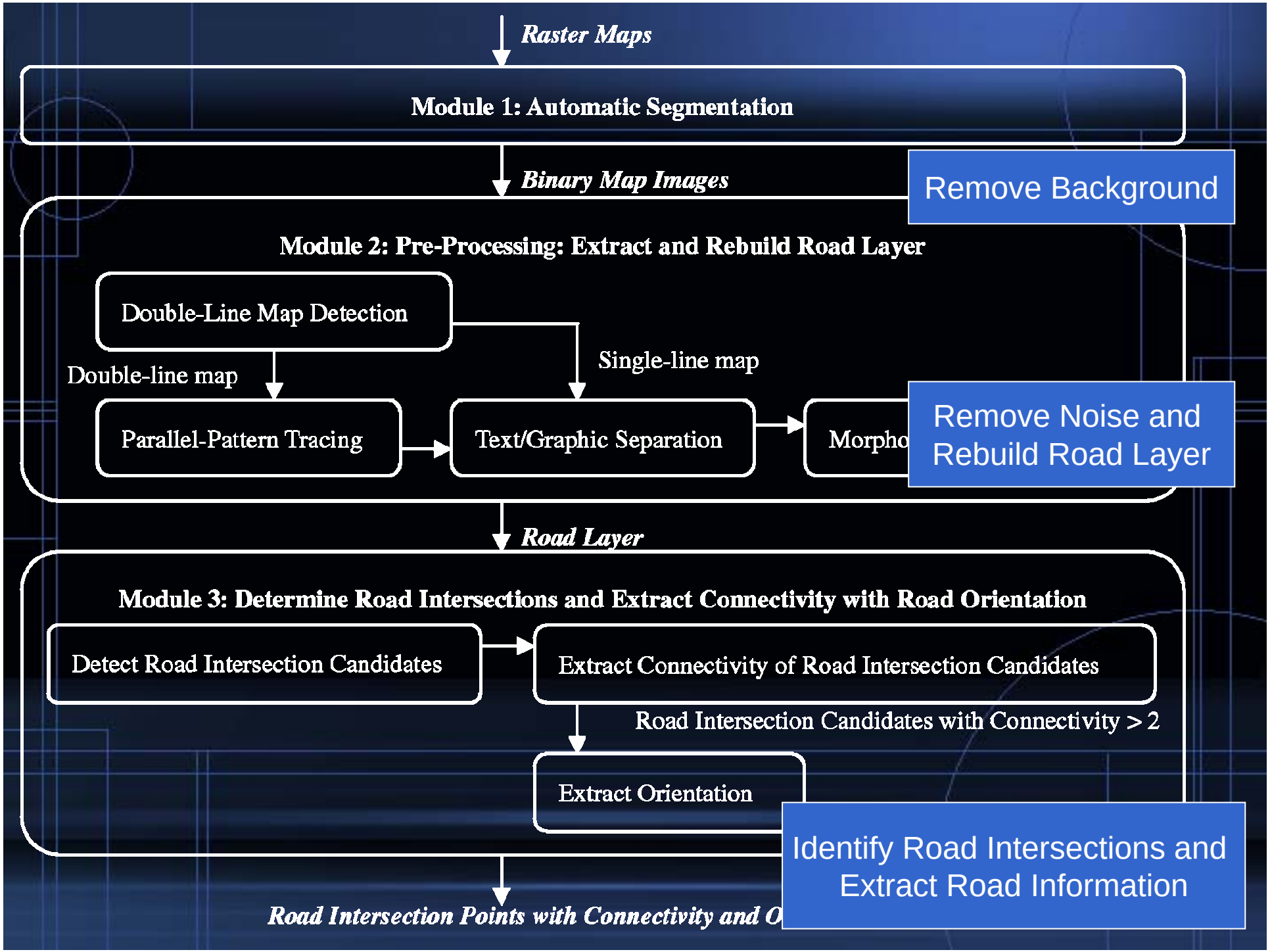
TIGER/Line Vector Data with Geo-coordinate Information

Outline

- Introduction and Motivation
- Approach and Algorithm
- Experimental Results
- Related Work
- Conclusion and Future Work

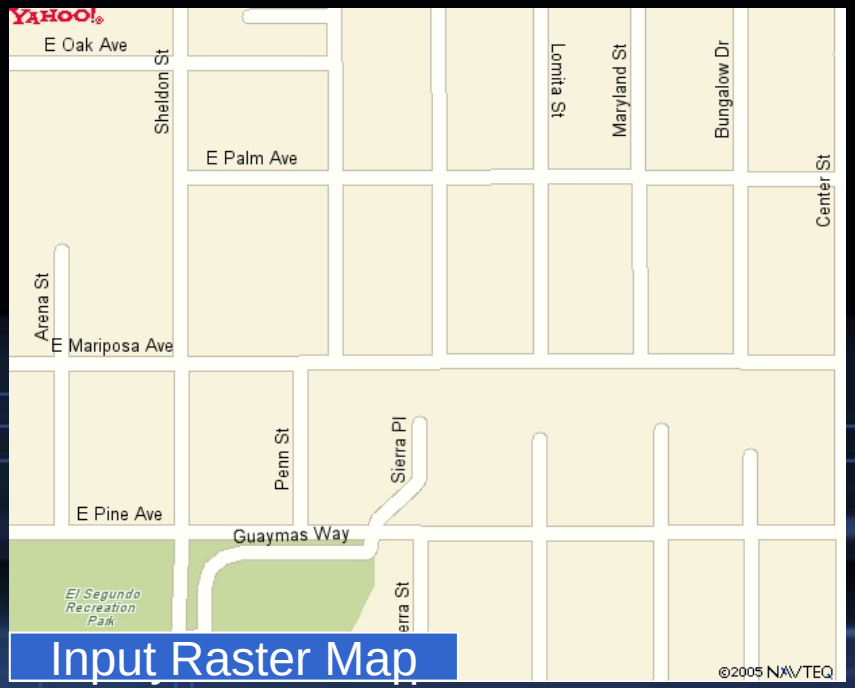
Approach and Algorithm

- For automatic road intersection extraction, we have to:
  - separate the road layer
  - extract road intersections


Raster Maps
Module 1: Automatic Segmentation
Binary Map Images
Remove Background
Module 2: Pre-Processing: Extract and Rebuild Road Layer
Double-Line Map Detection
Double-line map
Single-line map
Parallel-Pattern Tracing
Text/Graphic Separation
Morpho
Remove Noise and Rebuild Road Layer
Road Layer
Module 3: Determine Road Intersections and Extract Connectivity with Road Orientation
Detect Road Intersection Candidates
Extract Connectivity of Road Intersection Candidates
Road Intersection Candidates with Connectivity > 2
Extract Orientation
Identify Road Intersections and Extract Road Information
Road Intersection Points with Connectivity and O

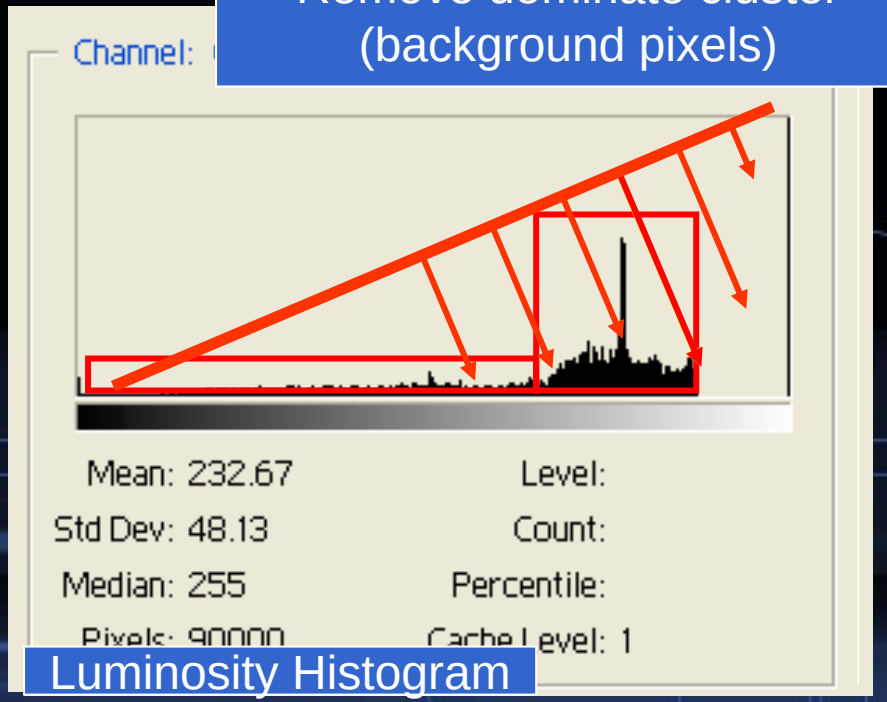
Remove Background

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



Input Raster Map

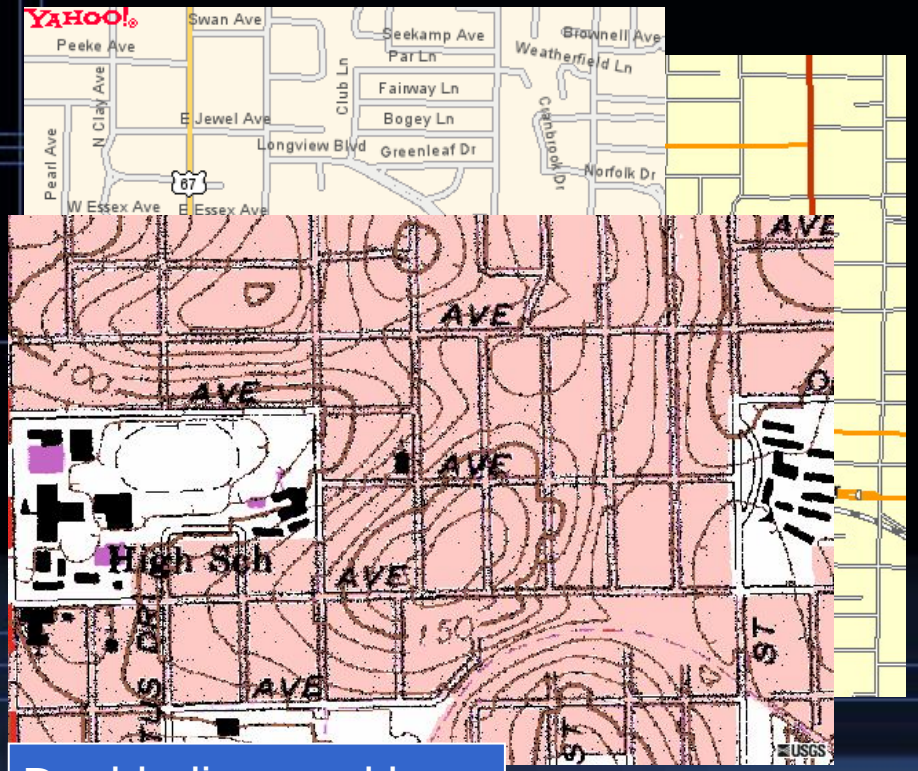
Remove dominate cluster (background pixels)



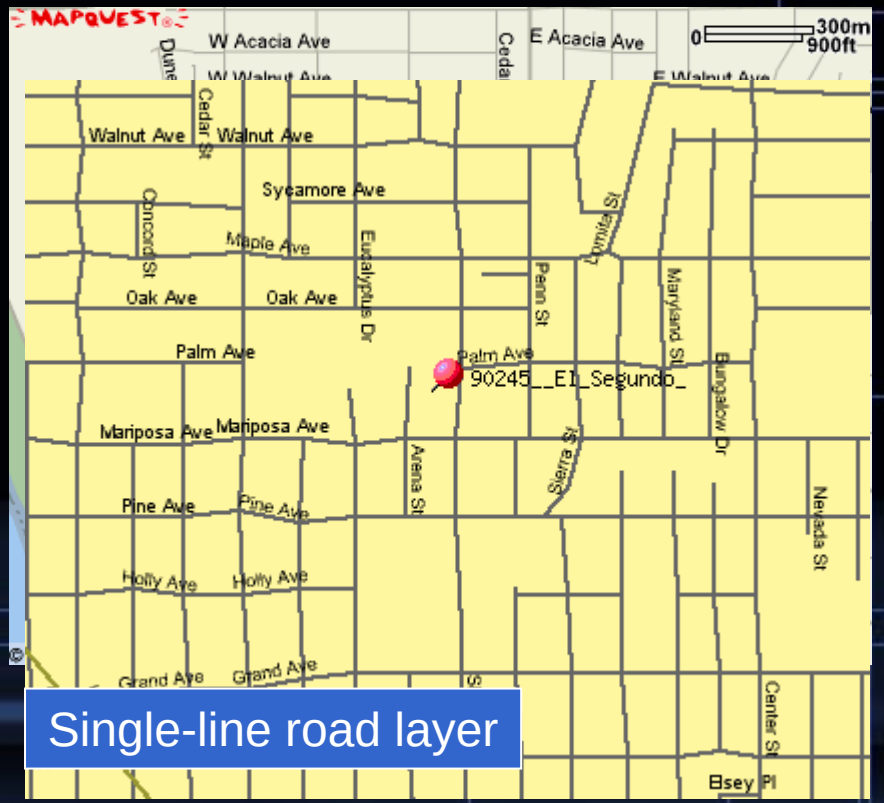
Luminosity Histogram

Remove Noise & Rebuild Road Layer

- Before we extract the intersections, we need to separate the road layer



Double-line road layer



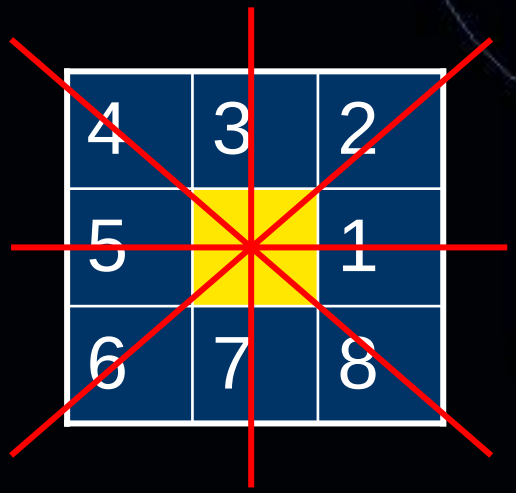
Single-line road layer

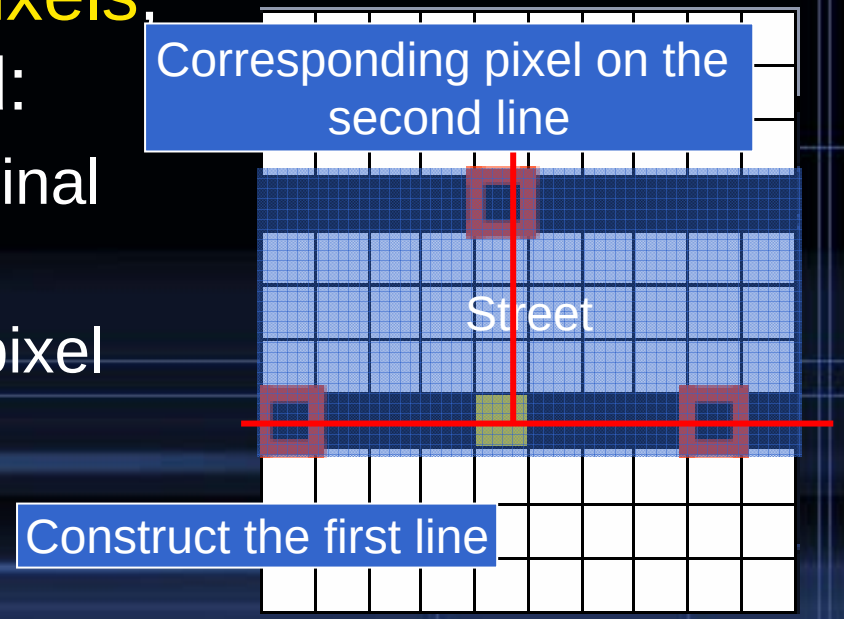
Remove Noise & Rebuild Road Layer

- Double-line road layers provide us more information to separate the road layer with other linear structure
- We utilize Parallel Pattern Tracing to find parallel road lines

Parallel Pattern Tracing

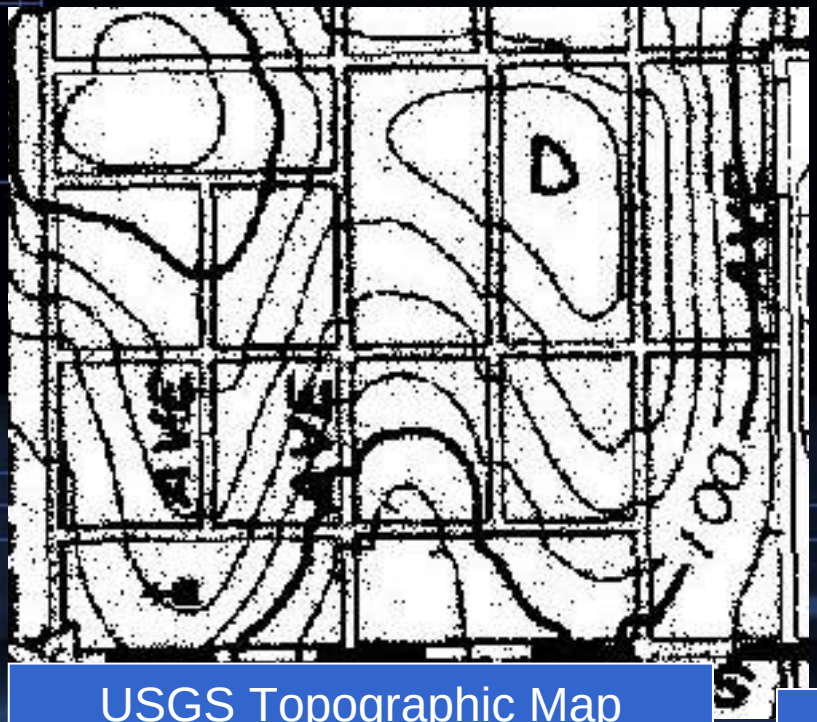
- Zoom in to pixel level:
  - 8 directions connect to one pixel
  - 4 possible straight lines
- If a pixel in on a double line layer with road width=3pixels, we should be able to find:
  - At least 1 pixel on the original road line
  - At least 1 corresponding pixel on the other road line



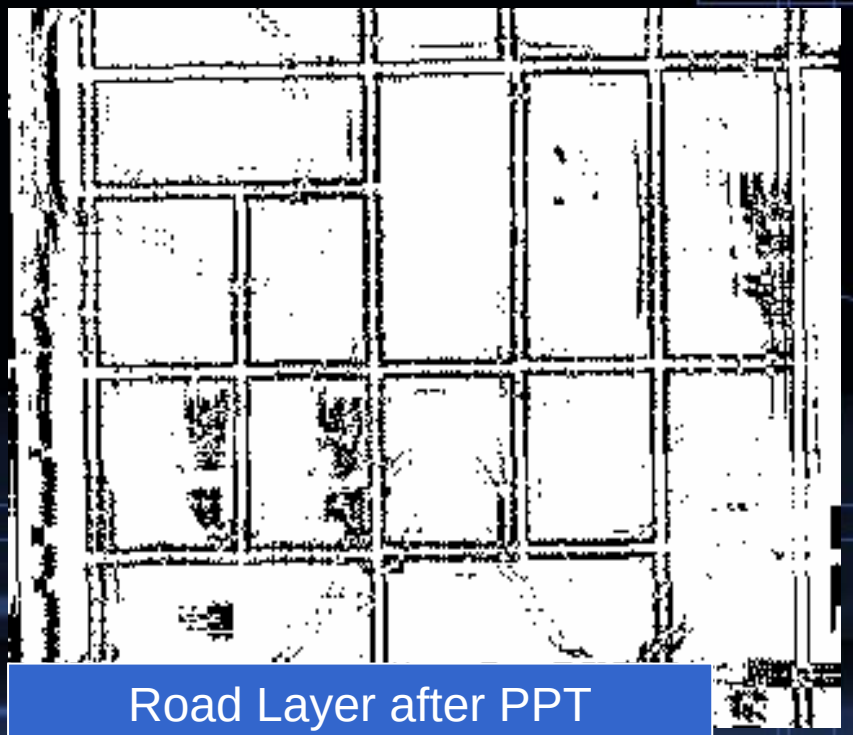


Parallel Pattern Tracing

- Detect the type of road layer, the road width
- Remove linear structures other than parallel roads



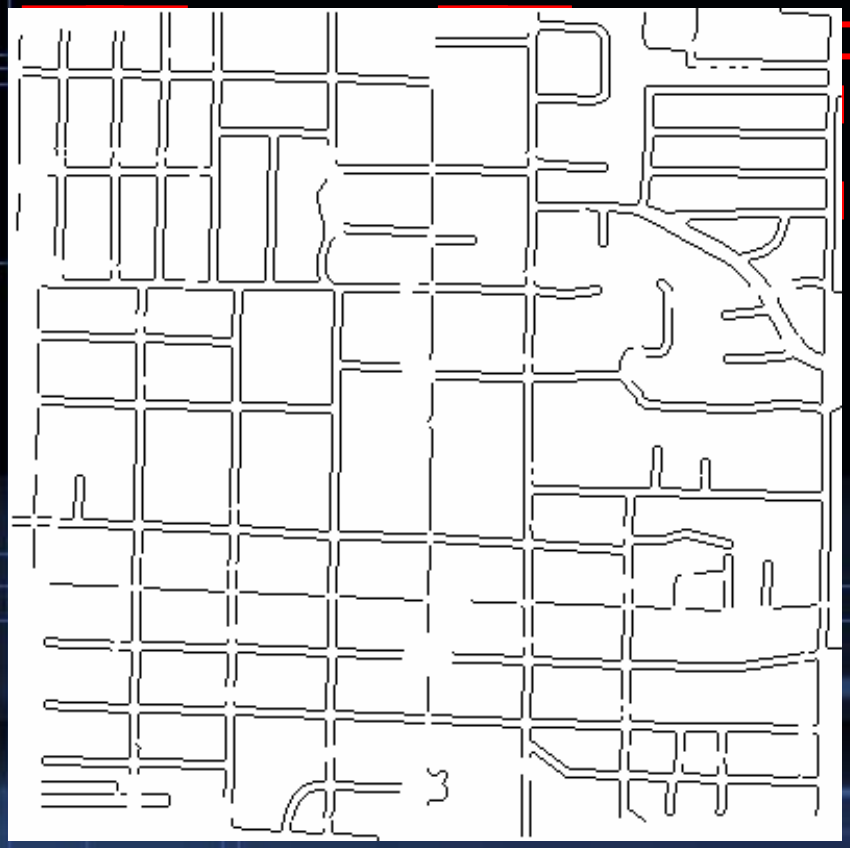
USGS Topographic Map



Road Layer after PPT

Remove Noise & Rebuild Road Layer

- Text/Graphics Separation (Cao et. al 2001)
  - Separate linear structures with other objects



Find small connected objects - character

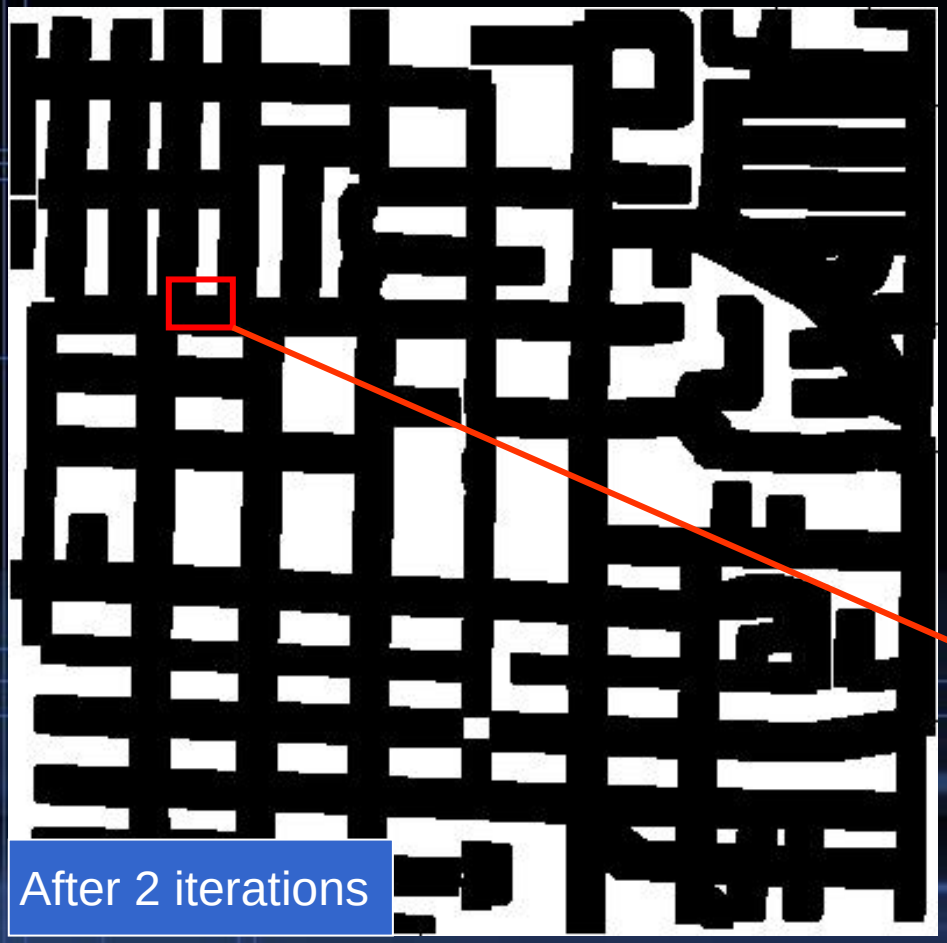
Grouping small connected objects - string

Remove small connected object groups

After the removal of
objects touching road lines,
the road network is broken

Rebuild Road Layer

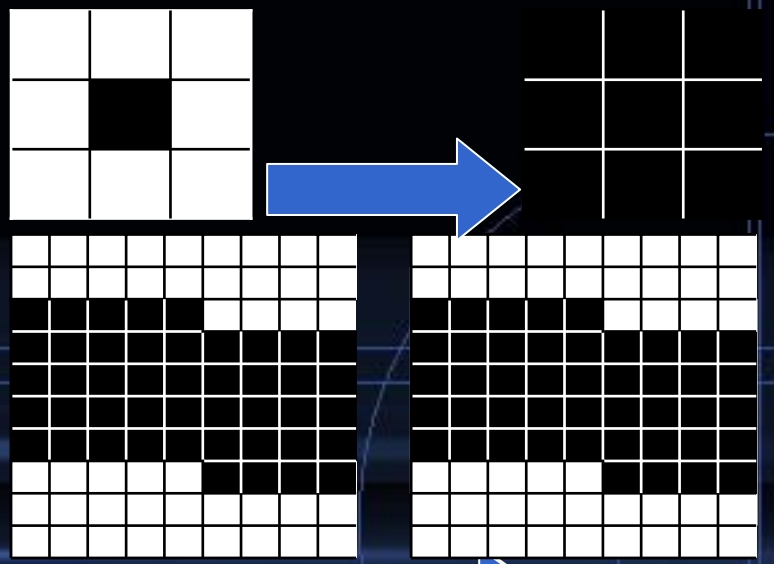
- General Dilation operator
  - Reconnect the broken road layer



After 2 iterations

Generalized Dilation

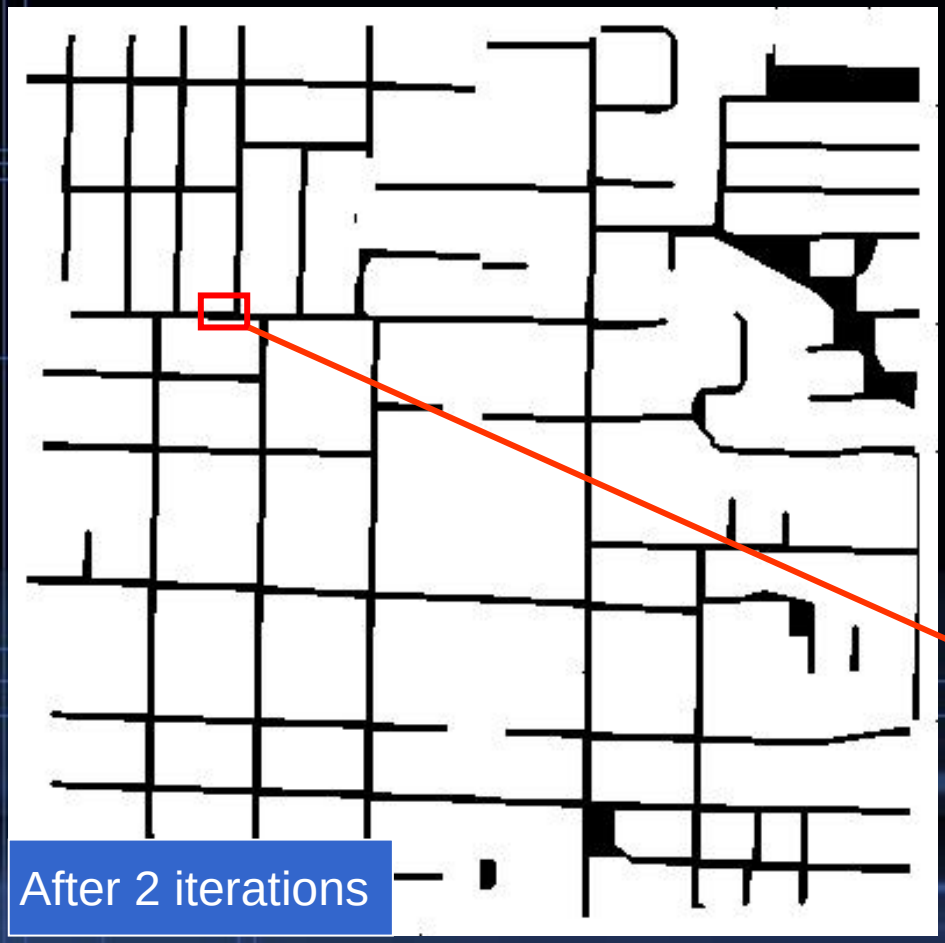
For every foreground pixel, fill up it's eight neighborhood pixels.



2nd iteration

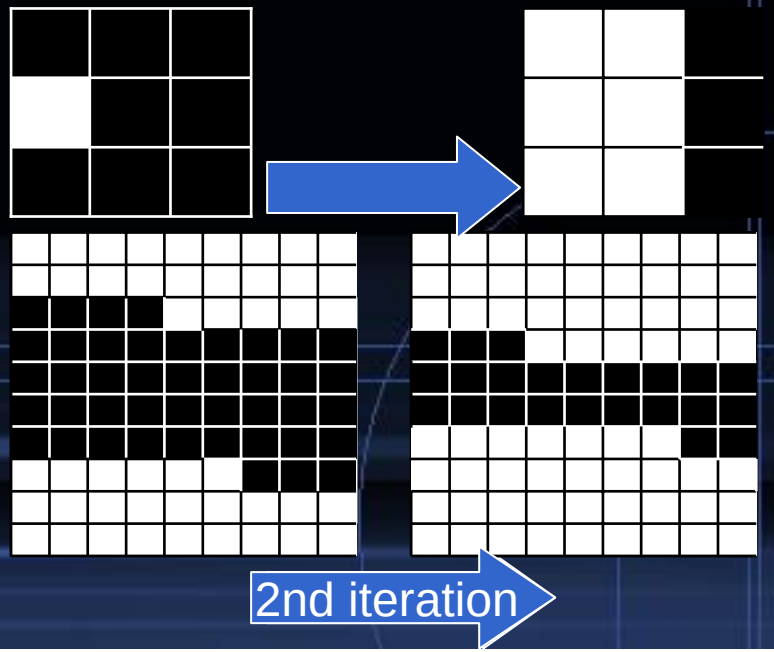
Rebuild Road Layer

- General Erosion operator
  - Thinner road lines and maintain the original orientation



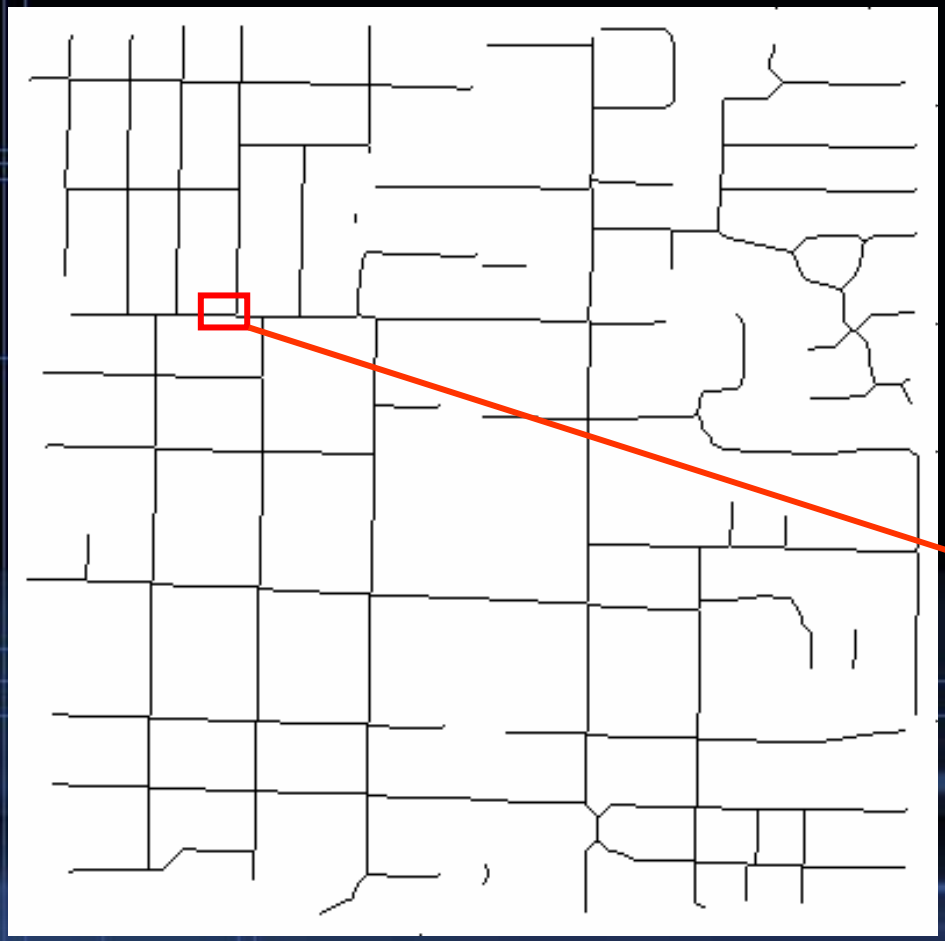
Generalized Erosion

For every foreground pixel, erase itself if any neighborhood pixel is white.



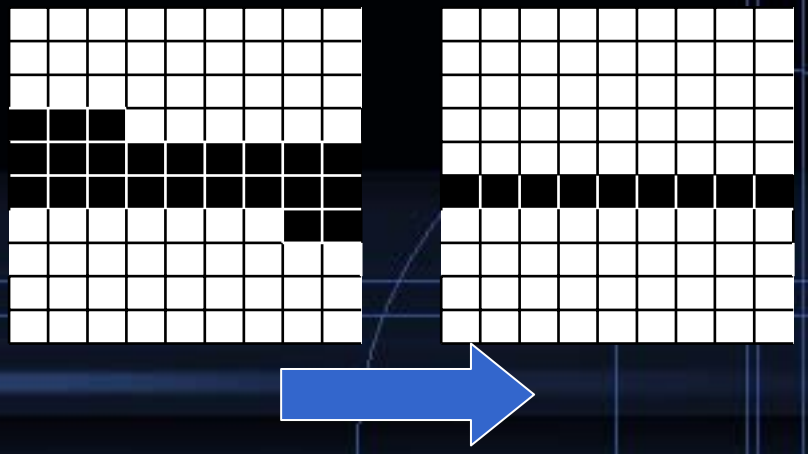
Rebuild Road Layer

- Thinning operator
  - Produce one pixel width road lines



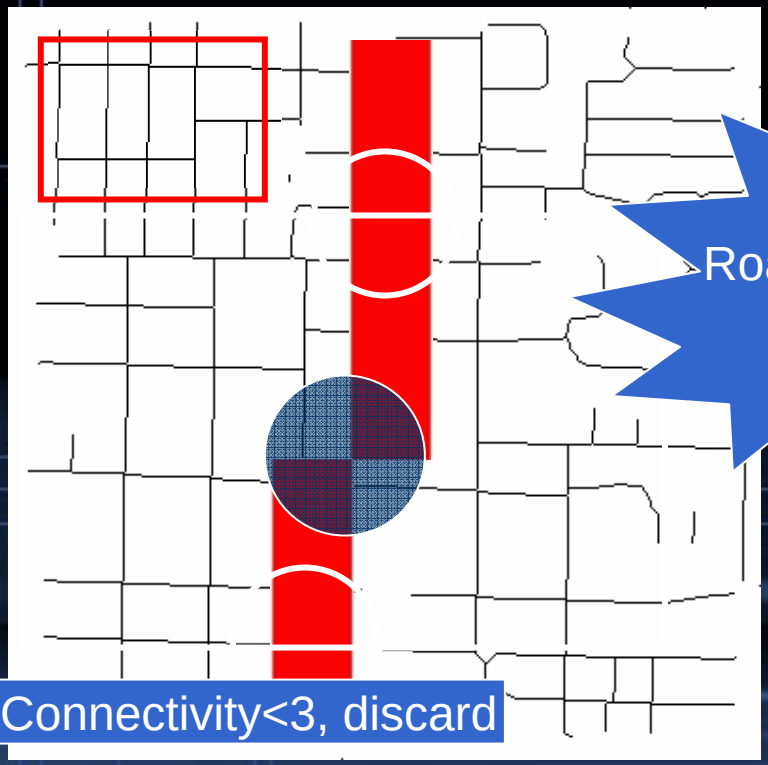
Thinning

Thinner each road line until they are all one pixel width.



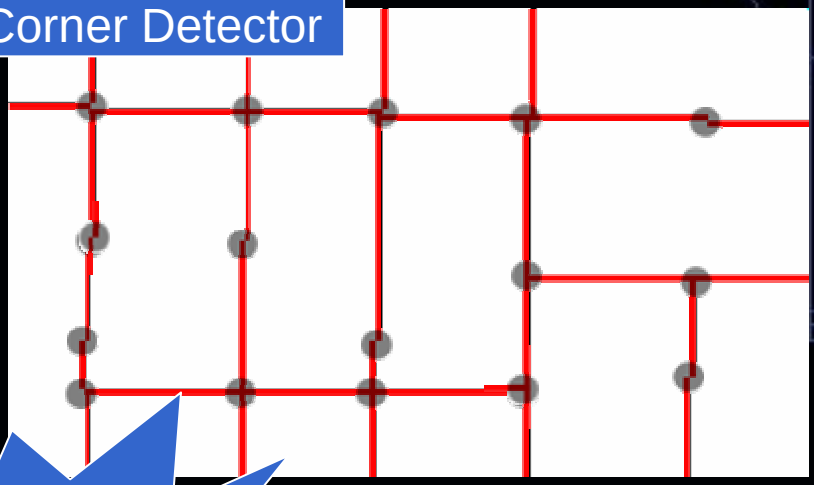
Identify Road Intersections and Extract Road Information

- Corner detector (OpenCV)
  - Find intersection candidates
- Compute the connectivity and orientation to determine correct intersections

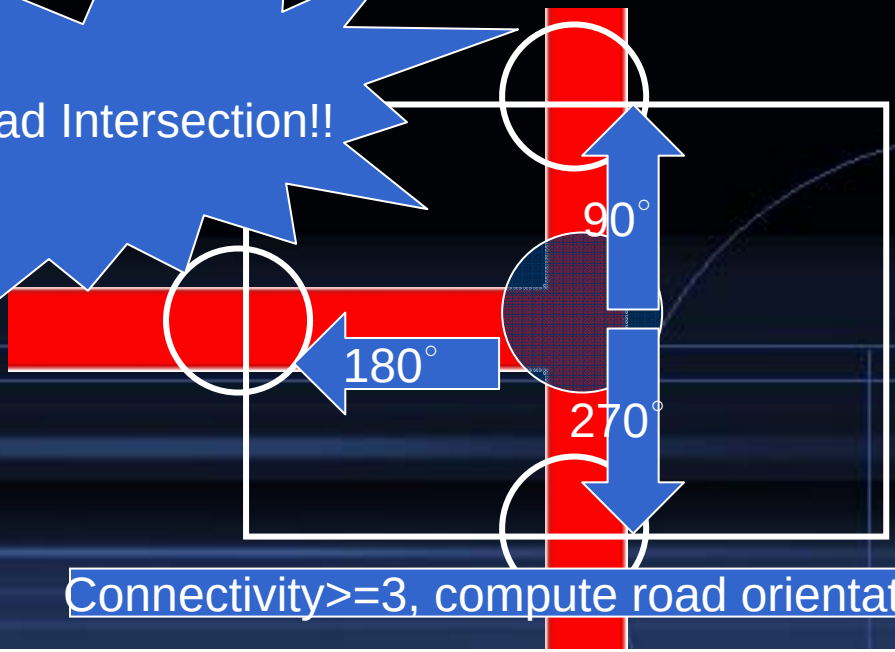


Connectivity<3, discard

Corner Detector



Road Intersection!!



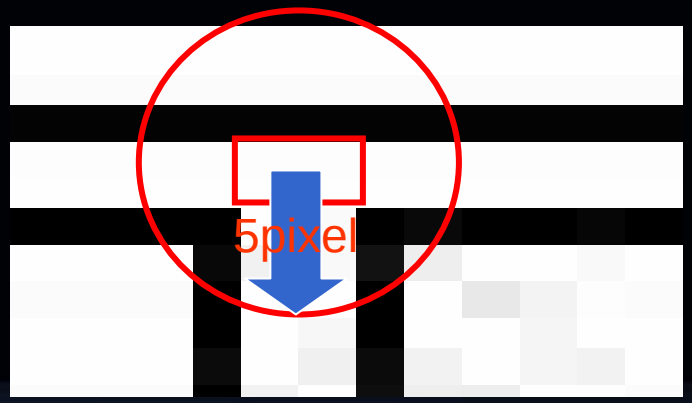
Connectivity>=3, compute road orientations

Outline

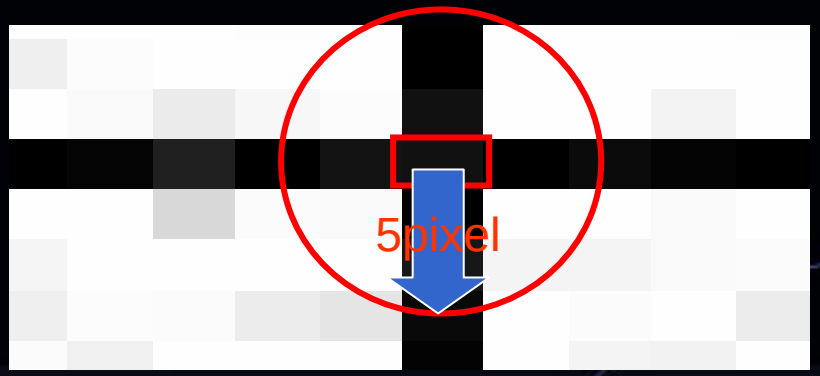
- Introduction and Motivation
- Approach and Algorithm
- Experimental Results
- Related Work
- Conclusion and Future Work

Experimental Results

- Correctly extracted intersection point:
  - Within 5pixels around an intersection point on the original map



Double-line road layer



Single-line road layer

Experimental Results

- CorrectINT - Correctly extracted road intersections
- AllExtractedINT - All extracted road intersections
- TotalINT – Actual road intersections on the raster map

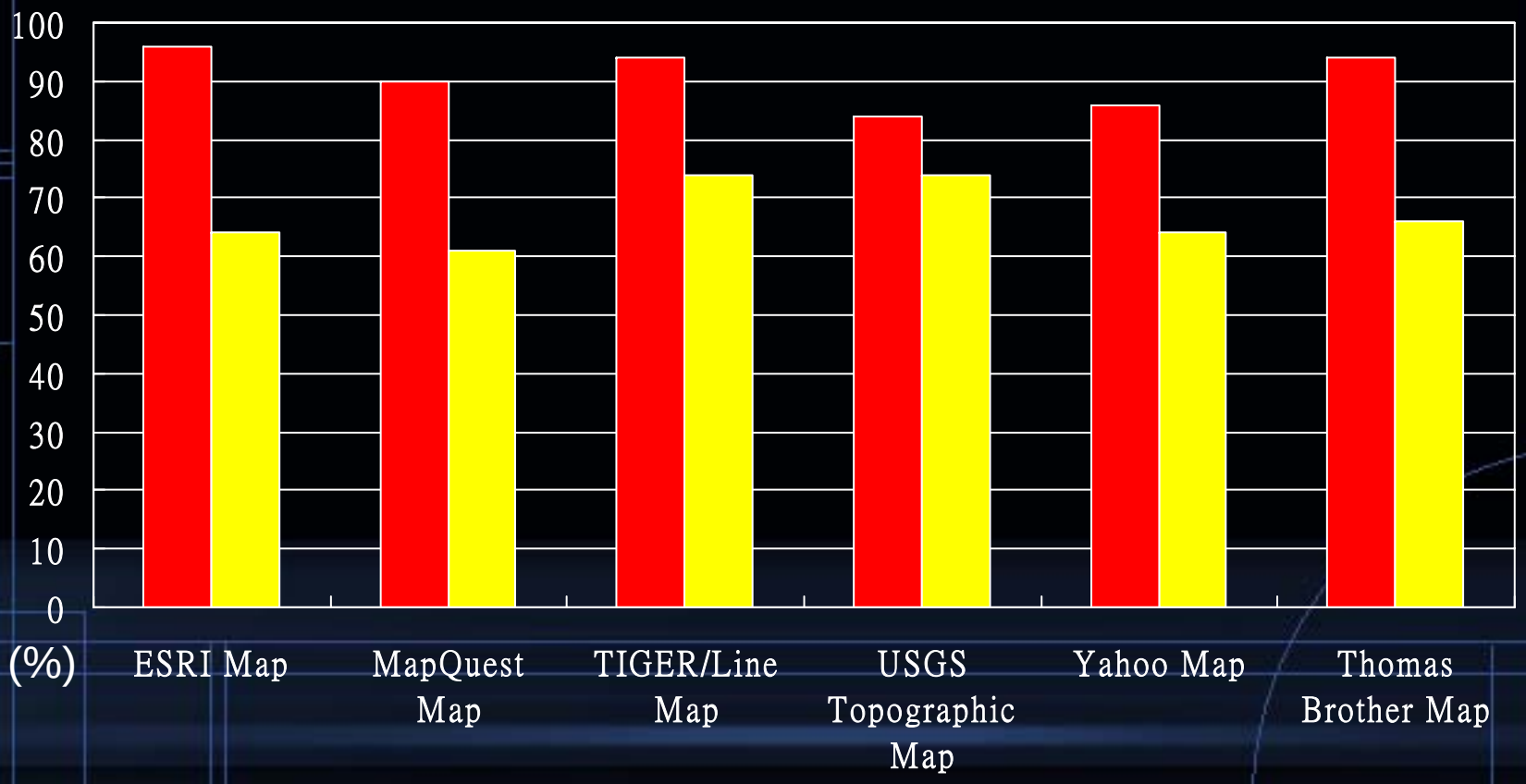
- Precision: P= CorrectINT / AllExtractedINT

- Recall: R= CorrectINT / TotalINT

- Positional accuracy:
  - The distance in pixels between the correctly extracted intersection and the corresponding intersection on the original map

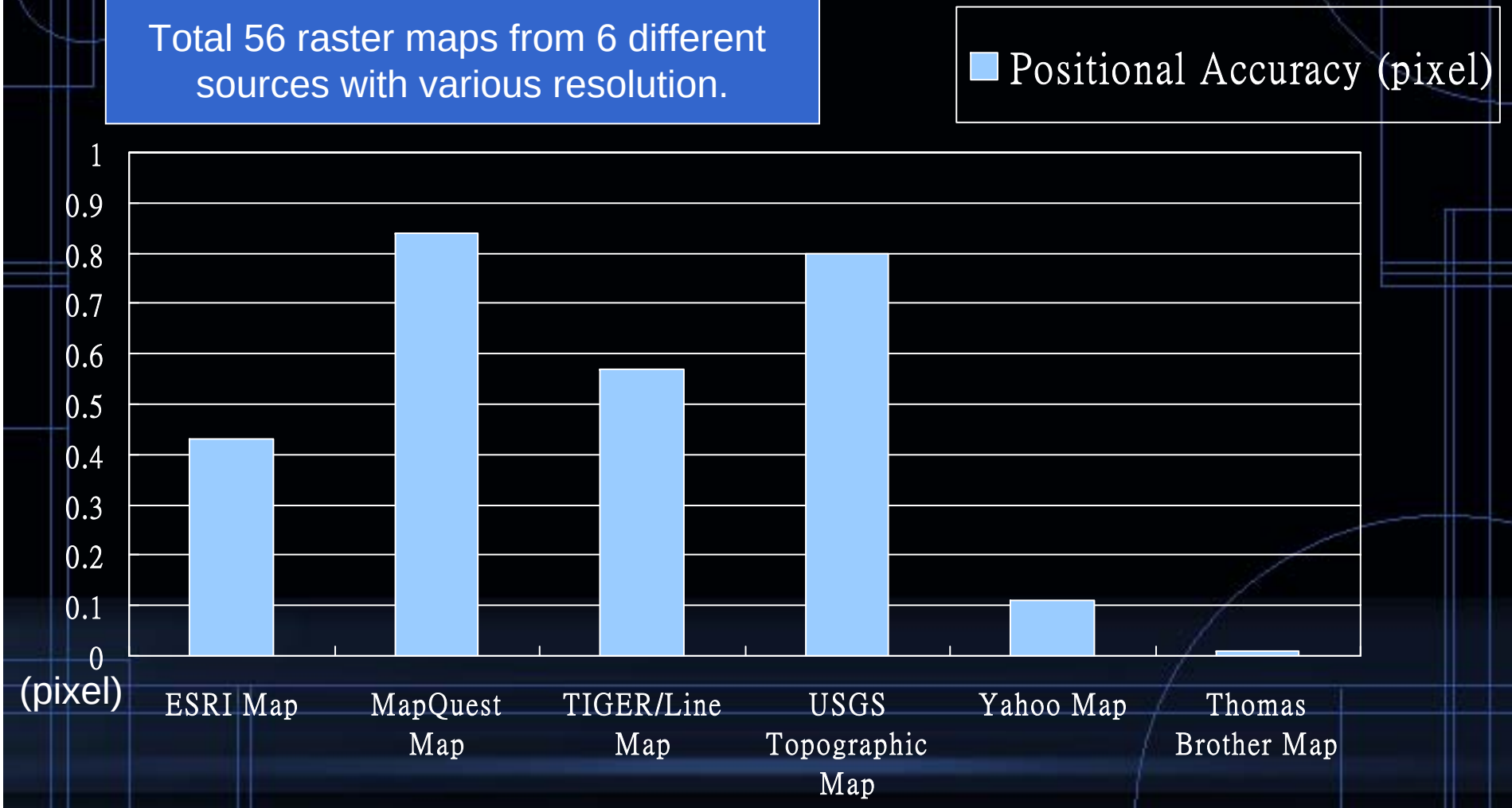
Experimental Results – Precision and Recall

Total 56 raster maps from 6 different sources with various resolution.



Experimental Results – Positional Accuracy

Total 56 raster maps from 6 different sources with various resolution.



Experimental Results - Performance

- Computation time:
  - Platform/Machine: Windows 2000 Server, Intel Xeon 1.8 GHZ Dual-Processor with 1 GB memory
  - 800x600 topographic map with resolution 2m/pixel: less than 1 minutes
  - Other simpler maps: less than 20 seconds

Outline

- Introduction and Motivation
- Approach and Algorithm
- Experimental Results
- Related Work
- Conclusion and Future Work

Related Work

- Contour line recognition from scanned topographic maps (Salvatore et. al 2001)
  - Use color classification to separate contour lines and use global topology information to reconstruct the broken lines
  - Require prior knowledge of the line color
- A legend-driven geographic symbol recognition system. (Samet et. al 1994)
  - Use the legend layer in a learning process to identify labels on the raster maps
  - Require legend layer and training

Related Work

- Automatic extraction of primitives for conflation of raster maps. (Habib et. al 1999)
  - Automatically extract primitives on raster maps
  - Require the input raster maps have only road layer and apply edge detector
- Verification-based approach for automated text and feature extraction from raster-scanned maps. (Myers et. Al 1996)
  - Use a verification based approach to extract data on raster maps
  - Require map specifications, legend layer and training

Outline

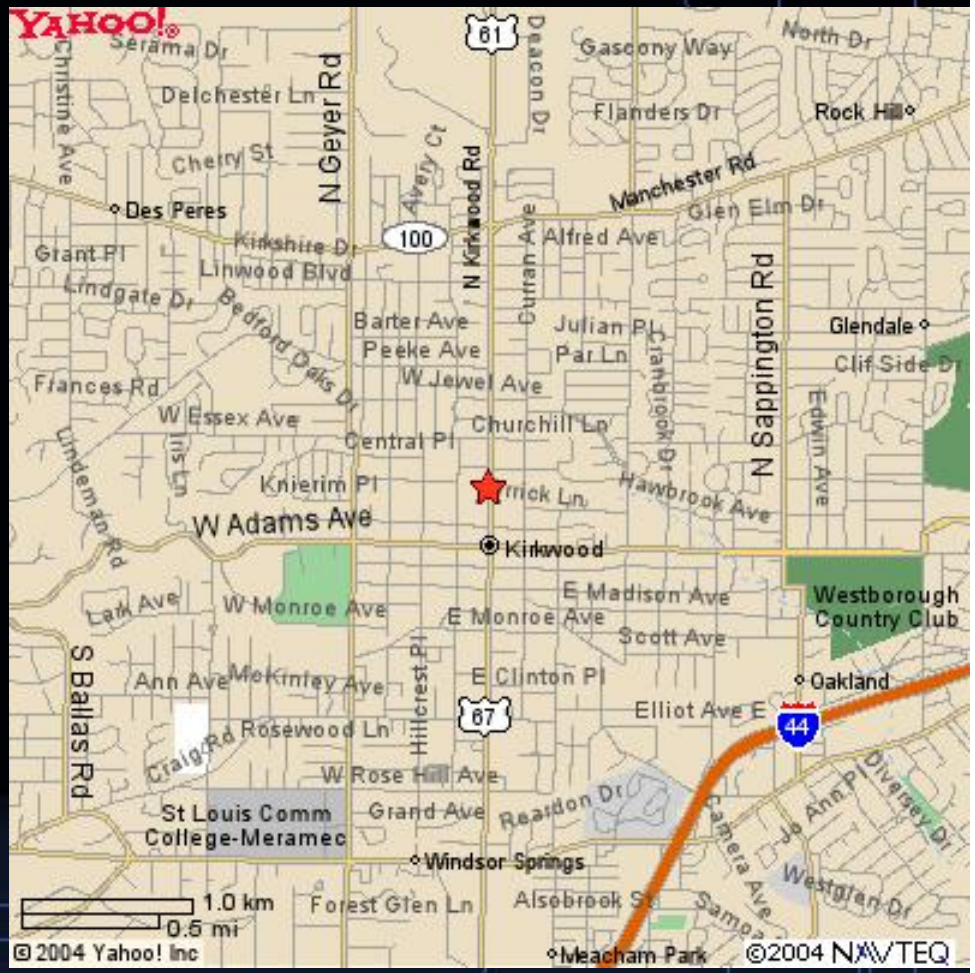
- Introduction and Motivation
- Approach and Algorithm
- Experimental Results
- Related Work
- Conclusion and Future Work

Conclusion and Future Work

- We achieved average 92% precision and 77% recall
  - Compared to 76% precision in previous work
  - Automatically extracting intersection points
  - Without prior information
- Efficient
- In our recent work Automatically Identifying and Georeferencing Street Maps on the Web (Sneha et al. 2005):
  - Found road intersections on automatically returned maps from image search engines
  - Identify the geocoordinates
  - Align the maps

Conclusion and Future Work

- Low-resolution maps:
  - many overlapped labels and lines
  - below average precision (66%) and low recall (27%)



Low-resolution Yahoo Map

Conclusion and Future Work

- Enhance the pre-processing modules to handle low-quality scanned map, more complex maps
- Combine Character Recognition module to “read” the map

Conclusion and Future Work

Thank YOU

Yao-Yi Chiang yaoyichi@isi.edu

University of Southern California
Information Sciences Institute