SEST-6577
Geographic Information Systems for Security Studies
Lecture 04 (Georeferencing and Vectorization)

Yuri M. Zhukov
Associate Professor
Georgetown University

September 26, 2024

What is **georeferencing**?

- assignment of geographic objects to geographic locations
- relation of map image to system of geographic coordinates on the ground

Figure 1: This is georeferencing

What is **vectorization**?

- generation of vector features from georeferenced raster images
- opposite is called rasterization (which is much easier)

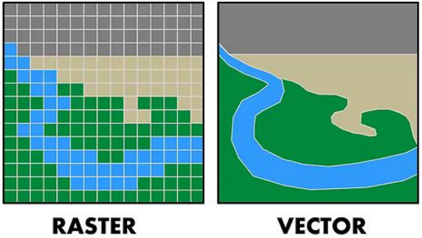

Figure 2: This is vectorization

Georeferencing

Why georeference?

- maps contain data you can't find anywhere else
- georeferencing allows us to
    - extract and preserve these data
    - combine map with other types of geospatial data
    - use these data to answer social, economic and political questions

Figure 3: NKVD jurisdictional borders

Overview

What is involved?

1. Obtain digital image of map (e.g. scan, web)
2. Select ground control points
3. Transform map to align with chosen coordinate system

Figure 4: Step 1

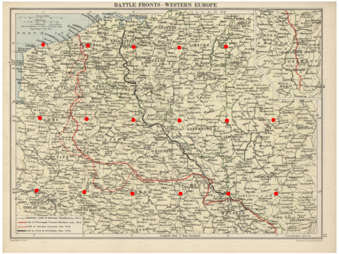

Figure 5: Step 2

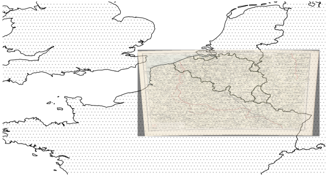

Figure 6: Step 3

What can we georeference?

- historical maps
- satellite and aerial photography
- administrative and military maps
- interesting maps you found online

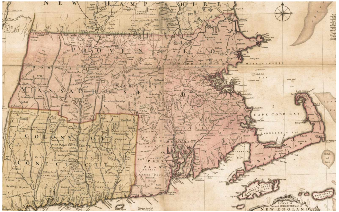

Figure 7: Massachusetts, 1755

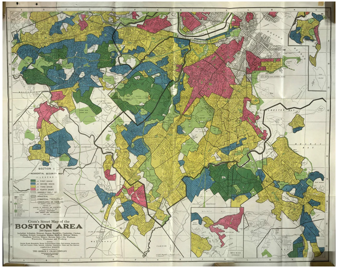

Figure 8: Boston, 1936

Figure 9: Boston, 1955

Challenges

- projection often unknown
- scale/resolution may be coarse
- distances/angles/shapes may be inaccurate (esp. in older maps)
- impossible to perfectly align historical maps with modern coordinate systems

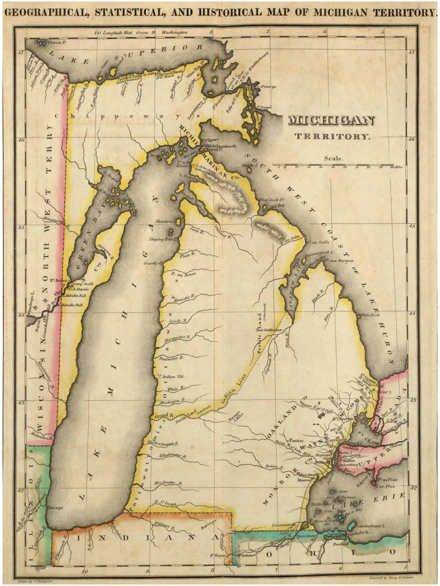

Figure 10: Close, but not quite

Examples of GCPs

- intersections of graticule lines (most reliable)
- landmarks of known location (e.g. buildings, crossroads, hills, cities)
- distinctive geographic features (e.g. coastal features, curves in rivers, borders)

Figure 11: Graticules

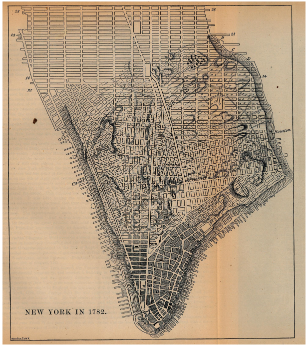

Figure 12: Intersections

Figure 13: Landmarks

Transformations

Transformation ("rubber sheeting")

- shift and warp the raster to spatially correct locations in original image
- apply mathematical algorithm to match source control points with target control points
- process changes distances, appearance of lines and shapes

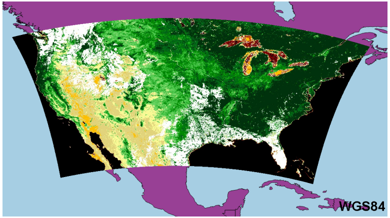

Figure 14: Transformed raster

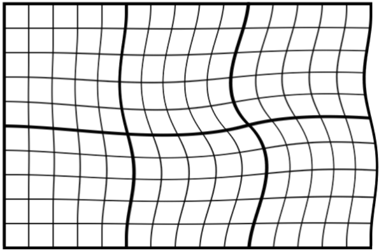

Figure 15: Rubber sheeting

Challenges

- transformation distorts original map image
- results sensitive to choice of transformation algorithm
- output only reliable within area confined by GCPs

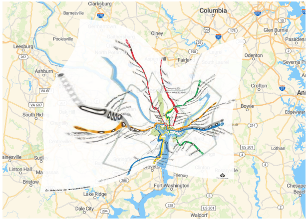

Figure 16: Distortion

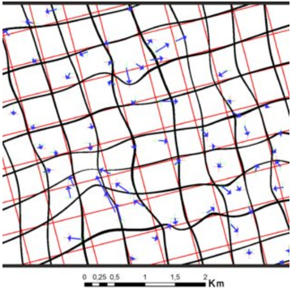

Figure 17: Algorithm

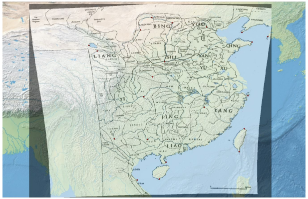

Figure 18: Range

Polynomial transformations

- uses a polynomial built on control points and a least-squares fitting algorithm
- optimized for global accuracy, not always local accuracy

Order

- $x_0 + x^1 + x^2 + x^3 + \cdots + x^k$, where k is *order of polynomial*
- higher order $\rightarrow$ able to correct more complex distortion
- but rarely need transformations > 3rd order

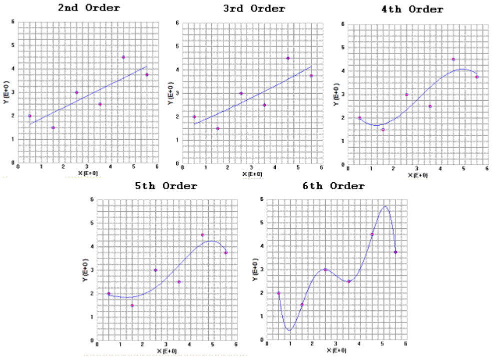

Figure 19: Higher order $\rightarrow$ closer fit

0. Zero-order polynomial
   - shifts raster location
   - used when raster is already georeferenced, but slightly mis-aligned
   - requires ≥ 1 control points
1. First-order (Affine) polynomial
   - shift/scale/rotate a raster
   - straight lines on input raster will remain straight
   - requires ≥ 3 control points

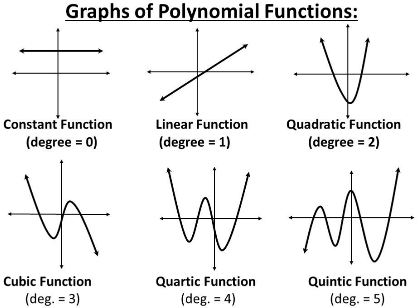

Figure 20: Examples in one dimension

2. Second-order polynomial
   - applies quadratic formula to calculate raster cell position
   - straight lines on input raster will be warped
   - requires ≥ 6 control points
3. Third-order polynomial
   - applies cubed formula
   - straight lines, margins on input raster will be warped
   - corrects more complex distortions
   - requires ≥ 10 control points

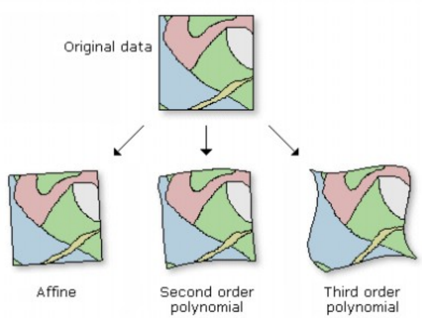

Figure 21: Examples in two dimensions

4. Projective transformation
   - linear rotation, translation
   - warps lines to keep them straight
   - useful for oblique imagery, scanned maps
   - requires ≥ 4 control points
5. Spline transformation
   - uses piecewise polynomial that maintains continuity between adjacent polynomials
   - optimized for local accuracy, not always global accuracy
   - minimal local error
   - requires ≥ 10 control points

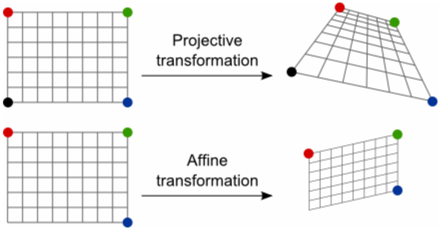

Figure 22: Projective vs. affine

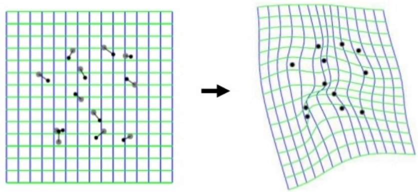

Figure 23: Spline

Vectorization

Why vectorize?

- vector is standard data structure for quantities of interest to public policy and social science (e.g. events, roads, administrative zones)
- smaller data size (usually)
- objects can have multiple attributes
- allows more sophisticated spatial analyses
- preserves quality at all scales

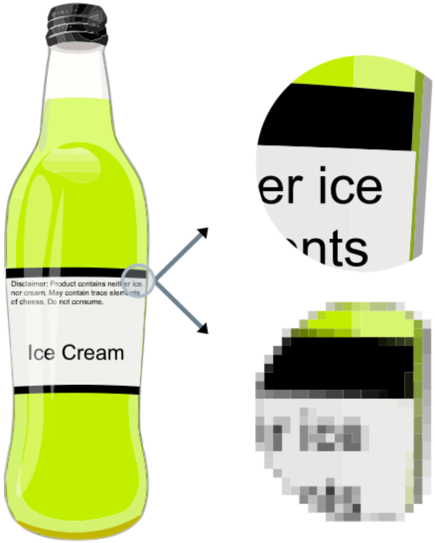

Figure 24: Enhance!

Options

Two ways to identify vector features

1. Image tracing (manual or automated)
2. Computer vision (machine learning)

Figure 25: Trace

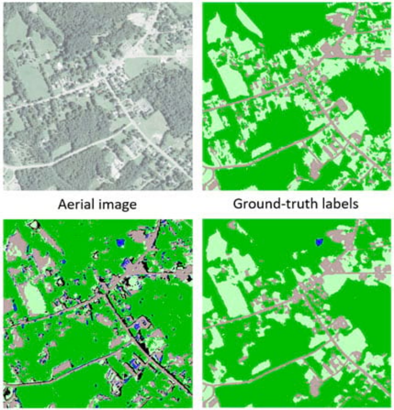

Figure 26: Learn

1. Image tracing
   - drawing over a raster image with vectors
   - *manual tracing*: tracing over the image by hand (using mouse or stylus)
   - *automated tracing*: use computer algorithm to detect features, redraw them as vector points, lines, polygons

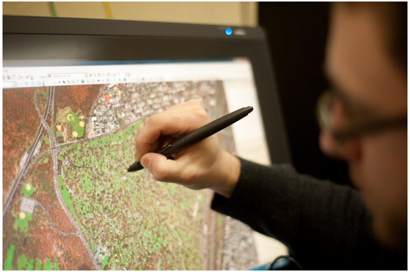

Figure 27: Manual

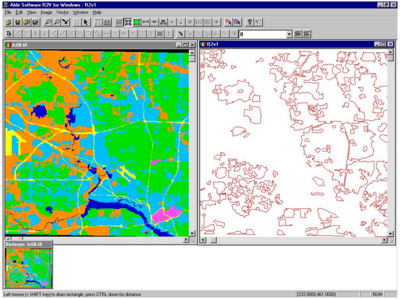

Figure 28: Automated

Manual tracing

Advantages

- can work with images of any quality
- better understanding of context/meaning
- produces fewer artifacts/superfluous features

Disadvantages

- slow, inefficient
- relatively imprecise/inconsistent
- subject to laziness/fatigue

Automated tracing

Advantages

- fast and efficient
- output is consistent, replicable

Disadvantages

- more sensitive to image quality
- can require extensive pre-processing/cleanup
- works best with fewer colors

2. Computer vision / deep learning
   - automated feature detection, extraction
   - system "learns" what is/isn't a feature through training data (e.g. examples of points, lines, polygons in raster images)
   - cross-validation of results to improve predictive fit
   - examples:
     - convolutional neural networks
     - recurrent neural networks
     - long short term memory models
     - transformer models

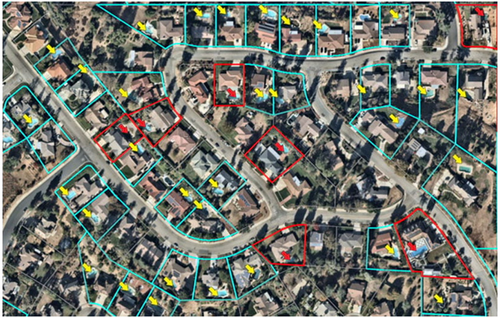

Figure 29: Which houses have pools

Machine learning

Advantages

- fast and efficient
- well-suited for large-scale tasks, where fixed rules lead to systematic errors

Disadvantages

- requires large volume of training data
- requires high-performance computing infrastructure, programming expertise
- same pre-/post-processing issues as automated tracing
- not (yet) available in standard GIS software

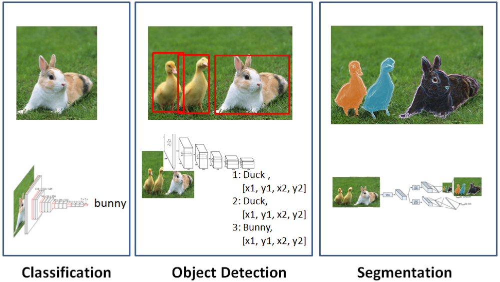

Figure 30: Computer vision tasks

Sources of error

Automated vectorization

1. Raster-to-point
   - all non-zero/non-null cells become points
2. Raster-to-line
   - trace positions of non-zero/ non-null pixels to identify polyline features
3. Raster-to-polygon
   - use groups of connected pixels with identical values to find areas of a raster
   - determine intersection points of area boundaries, generate lines

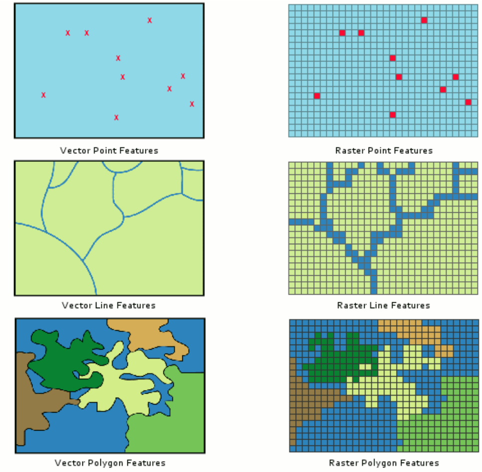

Figure 31: Usually not so seamless

Types of vectorization errors

1. *False positives*
   - identification of features where none exist
     (generates small/superfluous vertices that must be removed)
2. *False negatives*
   - failure to identify features where they exit
     (creates gaps, incomplete features)

How to reduce errors

1. *Pre-processing*
   - remove noise, unnecessary elements, colors
2. *Post-processing*
   - remove superfluous features, fill gaps, improve appearance

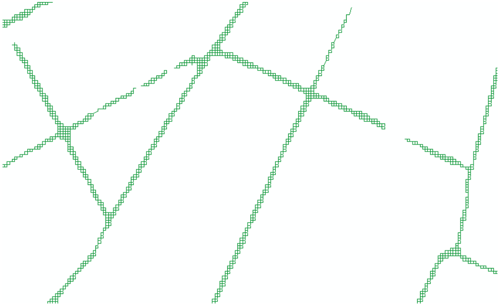

Figure 32: Vectorized "roads"

1. Pre-processing of rasters
   - *reclassification*: conversion from color/greyscale to binary
   - *thinning*: reduce thickness of features to a single, connected lines of pixels

Figure 33: Thinning example

2. Post-processing of vector features
   - *collapsing*: simplification by removal of spurious nodes, segments, closing of gaps
   - *smoothing*: generalization/ averaging to smooth pixelated appearance of output vectors

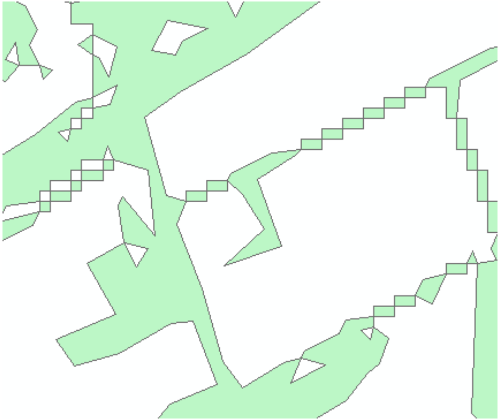

Figure 34: Lots to clean up