Using GRASS GIS and R

Assignment 2 for Spatial Statistics (STAT 946)

Adrian Waddell

University of Waterloo

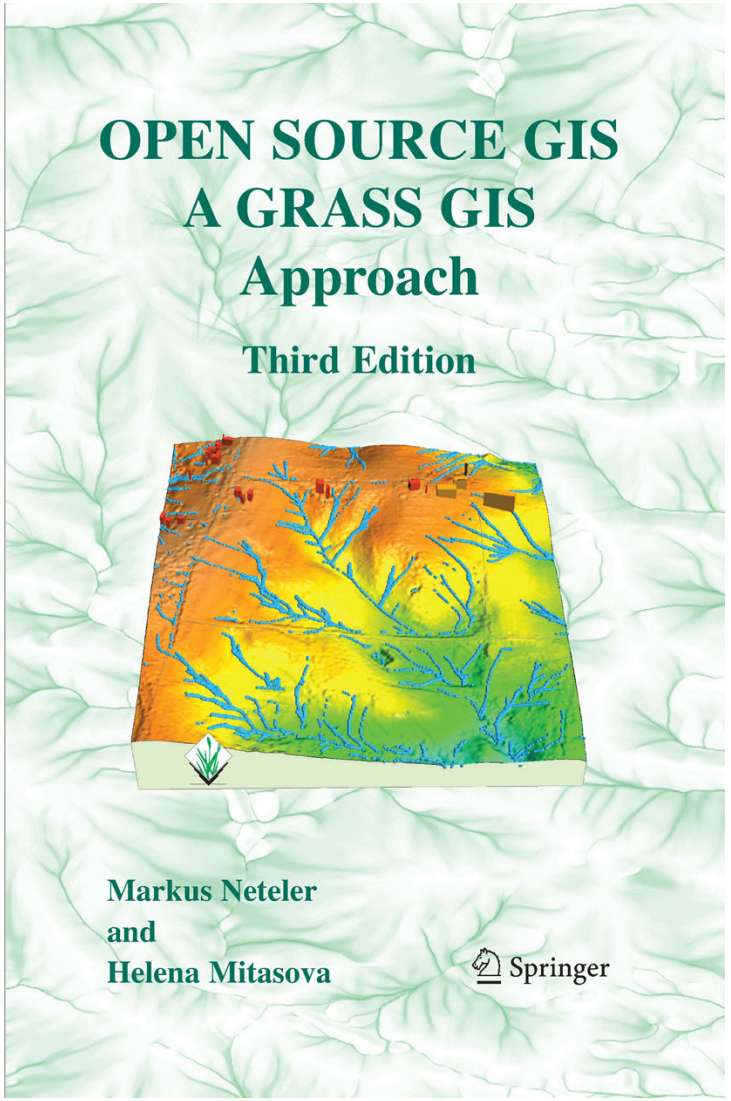
November 6, 2008

What is a GIS?

- **G**eographical **I**nformation **S**ystem
- does basically everything related to the handling and analysis of geographical data.
- GIS relates to a concept. Many GIS-Software tools exist: e.g. GRASS, ArcGis, GeoMedia, MapGuide, Quantum Gis.
- GRASS is open source and under the GNU General Public License
- GRASS has several GUIs and an API for R and other programming languages.

University of Waterloo

GRASS


OPEN SOURCE GIS
A GRASS GIS
Approach
Third Edition
Markus Neteler
and
Helena Mitasova
Springer

GIS Concepts

- Two types to represent spatial data:
  - field representation (image): raster data
  - geometrical objects (lines, points, polygons): vector data
- Make sure for each map, what represents what, e.g. city $\Rightarrow$ point or a polygon, river $\Rightarrow$ line or polygon, etc...
- raster data: pixel is 2D and a voxel is 3D. Often used for analysis, modeling and image processing.
  - \+ Simplicity
  - – not efficient for data dependent on lines and boundaries

University of Waterloo

GIS Concepts

- Two types to represent spatial data:
  - field representation (image): raster data
  - geometrical objects (lines, points, polygons): vector data
- Make sure for each map, what represents what, e.g. city $\Rightarrow$ point or a polygon, river $\Rightarrow$ line or polygon, etc...
- raster data: pixel is 2D and a voxel is 3D. Often used for analysis, modeling and image processing.
  - \+ Simplicity
  - – not efficient for data dependent on lines and boundaries

GIS Concepts

- Two types to represent spatial data:
  - field representation (image): raster data
  - geometrical objects (lines, points, polygons): vector data
- Make sure for each map, what represents what, e.g. city $\Rightarrow$ point or a polygon, river $\Rightarrow$ line or polygon, etc...
- raster data: pixel is 2D and a voxel is 3D. Often used for analysis, modeling and image processing.
  - \+ Simplicity
  - – not efficient for data dependent on lines and boundaries

University of Waterloo

Vector data model

- arc: non-intersecting line
- line: a series of (x, y) or (x, y, z) points
- nodes: endpoints of an arc
- vertices: points along a line
- segment: two consecutive points
- Arcs form higher level map features like areas and polygons.
- centroid: center of area
- area: closed by arc edges and a centroid
- faces: a 3D polygon
- kernel: 3D centroid
- 3D Volume: closed set of faces with a kernel
- some GIS Software support polynomials and splines to describe vertices

Organization of geographical data

- use of thematic maps (layers)
- save attributes in databases
- possible to generate new maps from several maps
- Some GIS have interface to Web-Geodata (Google Earth, NASA World Wind)

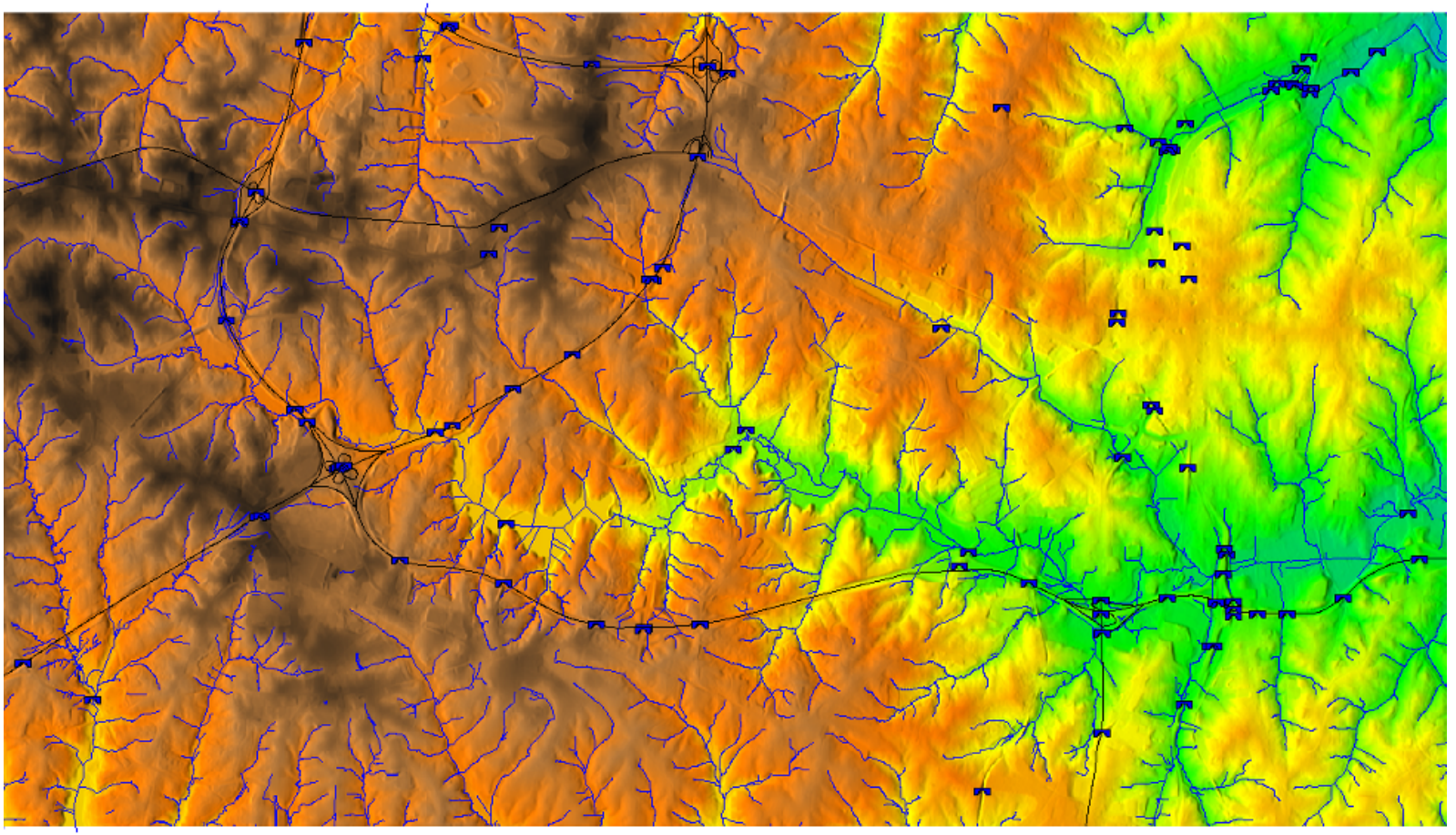
Map projections and coordinate systems

- Shape of Earth is usually modelled as an ellipsoid (spheroid), commonly used ellipsoids models are
  - Clarke 1866 for North America
  - Bessel 1841 for European Countries
  - WGS 1984 for the whole world
- also sometimes a geoid is used (equipotential surface of earths gravity)
- Geodic or map datum
  - Horizontal datum: (x, y) origin
  - Vertical datum: z origin (like mean sea level)
- Map projection As defined in class
- EPSG maintains a list with all common projection definition. Therefore always look if there is a EPSG projection code for your data (used in GRASS and R). www.epsg.org

University of Waterloo

Getting started with GRASS

- Install GRASS, download example data (North Carolina)
- Go through sample session in book p.29



University of Waterloo

Getting started with GRASS, cont.

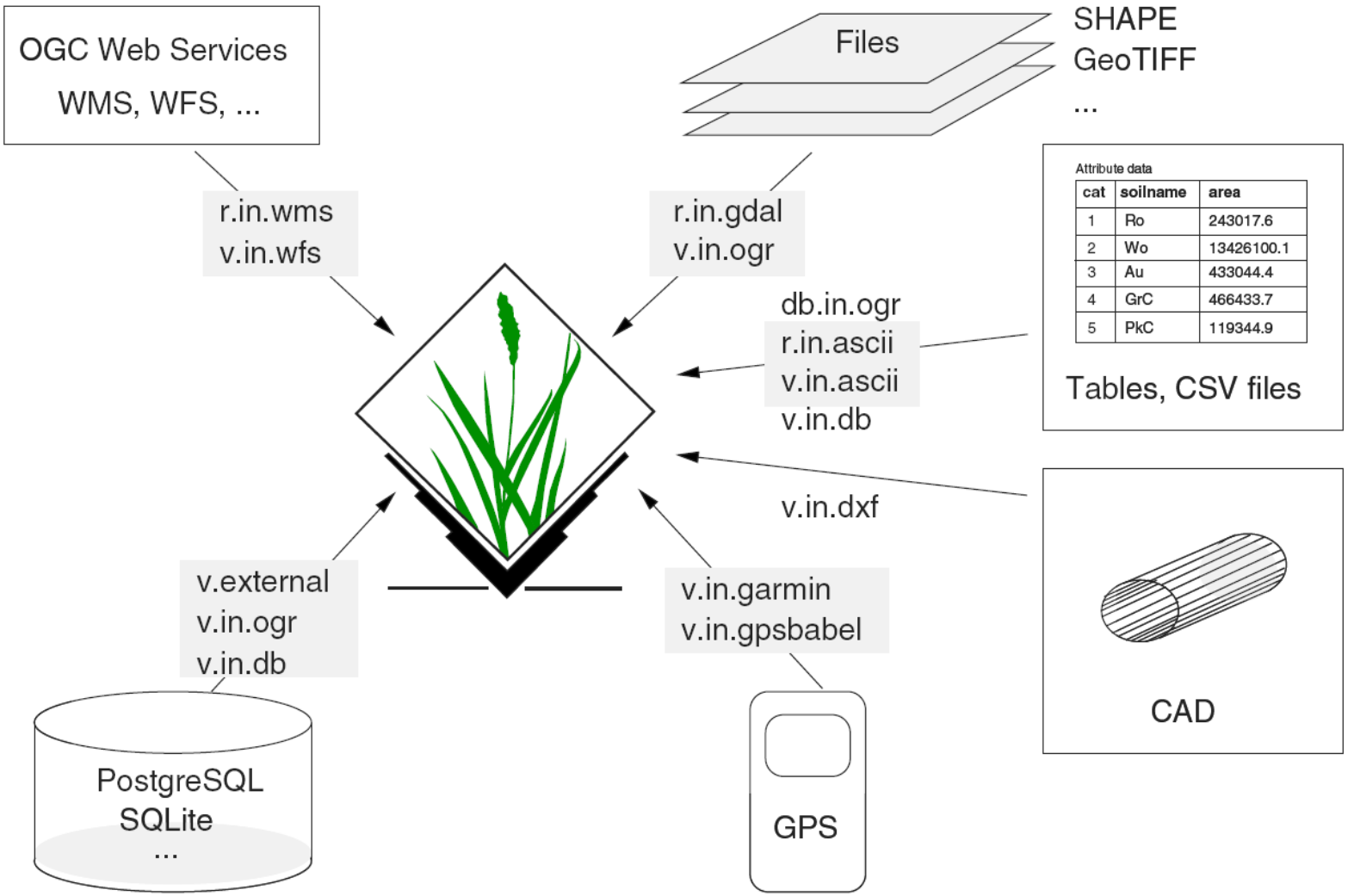
- It is important to know your data well (projection, structure, etc...)
- QGIS is a GUI with GRASS Interface. (www.qgis.org)
- A list of free GIS programs and free Maps: www.freegis.org
- GDAL is a library with data format definitions (import,export, raster, vector).
- GDAL is also available in R.

University of Waterloo

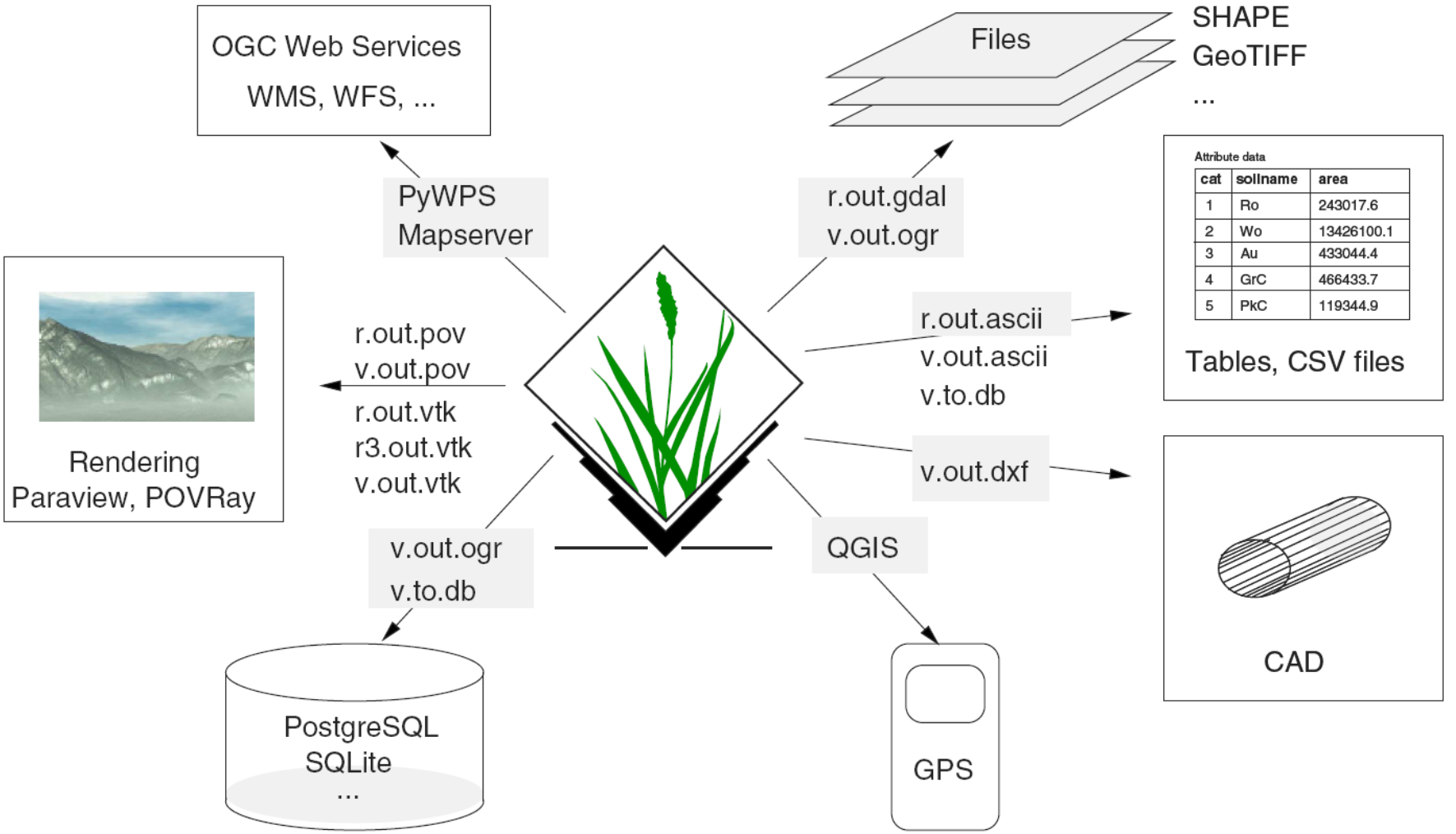
GRASS function classes

prefix	function class	type of command
d.	display	graphical output
db.	database	database management
g.	general	general file operations
i.	imagery	image processing
m.	misc	miscellaneous commands
ps.	postscript	map creation in Postscript format
r.	raster	2D raster data processing
r3.	3D raster	3D raster data processing
v.	vector	2D and 3D vector data processing

Data handling: import

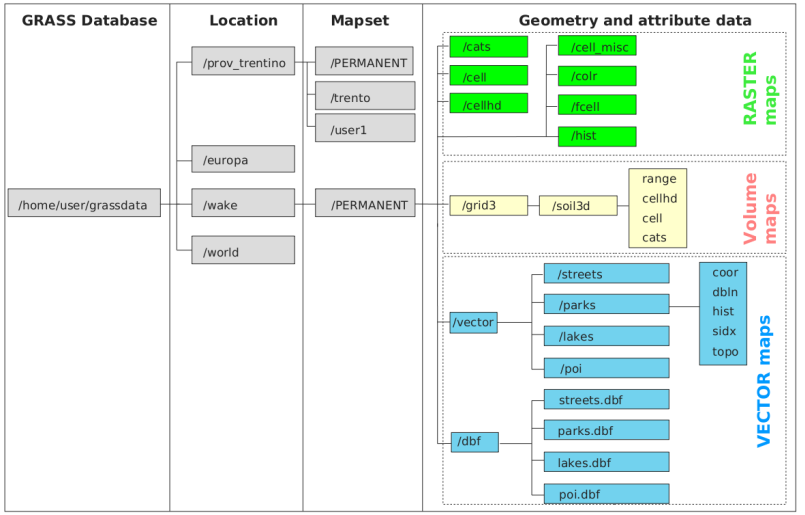


Data handling: export



GRASS internal Maps organisation

- GRASS physical data structure



- Each LOCATION is defined by its coordinate system, map projection and geographical boundaries
- MAPSETS are used to subdivide the project into different topics, subregions, or as workspaces for individual users

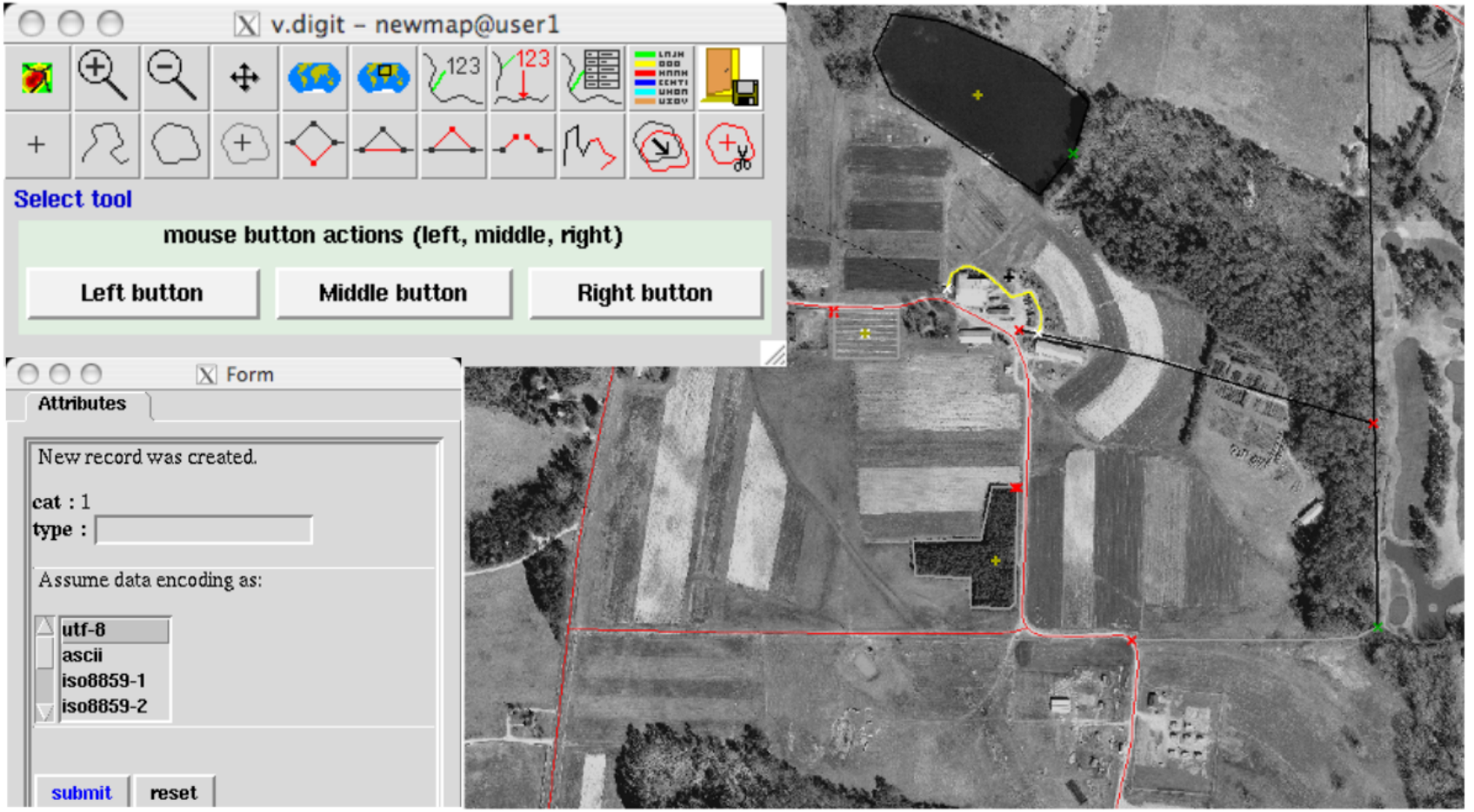
University of Waterloo

Import data

- Transform projection of vector data with ogr2ogr
- import vector data with v.in.ogr

GRASS: Digitize data

- Turn raster data into vector data



Visualize 3d

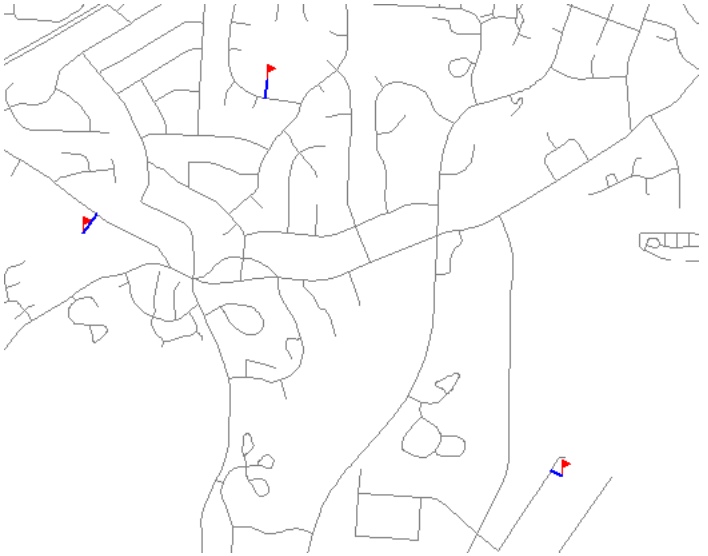
- Raster data is used to store elevation data (elevation matrix)
- display 3d raster data with `nviz elevation_ned_30m`

Vectormap queries and statistics

- Get information about displayed map d.what.vect
- Get average elevation for all zip codes v.rast.stats
- other methods are limited, use R or gstat

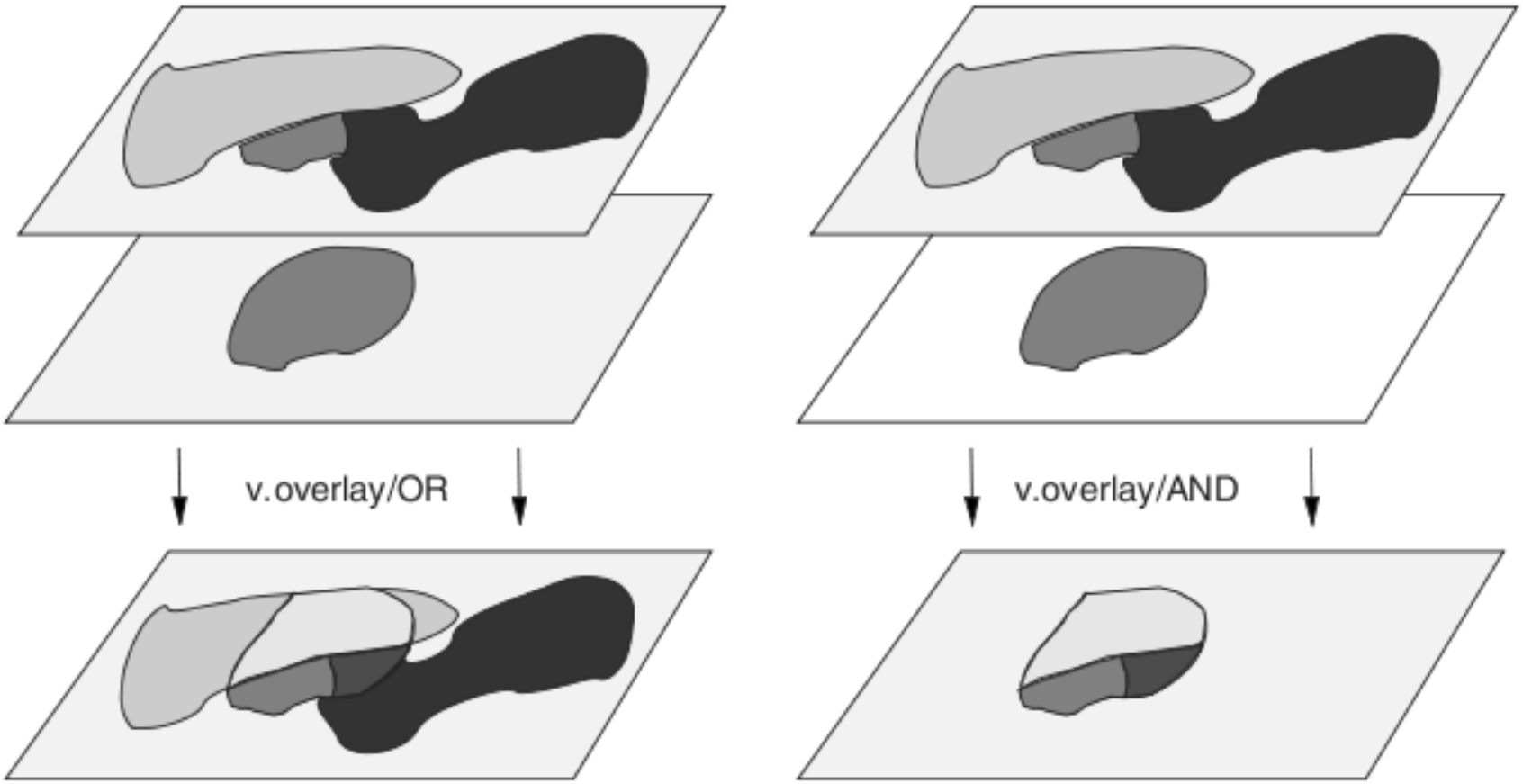
Geometry analysis

- Analysis of adjacency and common boundaries
- Analysis of distance, e.g. distance from schools to closest road $\rightarrow$ generate new vector data (in blue), code p.202



- Select Poins in Polygon

Merging vector maps, p/206



Vector network analysis

- Vector network maintenance: v.net;
- Shortest path: d.path and v.net.path;
- Traveling salesman (round trip): v.net.salesman;
- Allocation of sources (create subnetworks, e.g. police station zones): v.net.alloc;
- Minimum Steiner trees (star-like connections, e.g. broadband cable con nections at mimimum costs): v.net.steiner;
- Iso-distances (from centers): v.net.iso.

GRASS and R

- Geostatistical analysis with GRASS isn't covered in the Neteler Book. He uses gstat and R in connection with GRASS to do variogram, krigin, etc...
- The R/GRASS interface is integrated into the sp "spatial" classes as extension spgrass6
- In R: `install.packages("spgrass6", "gstat", dependencies = TRUE)`
- To understand sp objects in R, read the book "Applied Spatial Data Analysis with R", Springer

University of Waterloo

GRASS and R, cont

- Start GRASS
- in GRASS shell environment, run R
- in R load `spgrass6`, i.e. `library(spgrass6),library(spatial)`
- have functions `readVECT6,gmeta6,readRast6`
- run GRASS commands within R with `system`

Conclusions

- GRASS and (GIS in general) have powerful tools
- however data structure is very complex. In addition need knowledge of databases and SQL.
- I wasn't able to import the Swiss data (because of the projection system)
- several functions didn't work, segmentation fault
- R has with the library sp similar capabilities for visualizing
- sp data structure is also though to handle
- Finally, working with geographical data needs experience

University of Waterloo

End

THANK YOU

University of Waterloo