The GRASS GIS software

GIS Seminar

Politecnico di Milano

Polo Regionale di Como

M. Neteler
neteler at itc it
http://mpa.itc.it
http://grass.itc.it

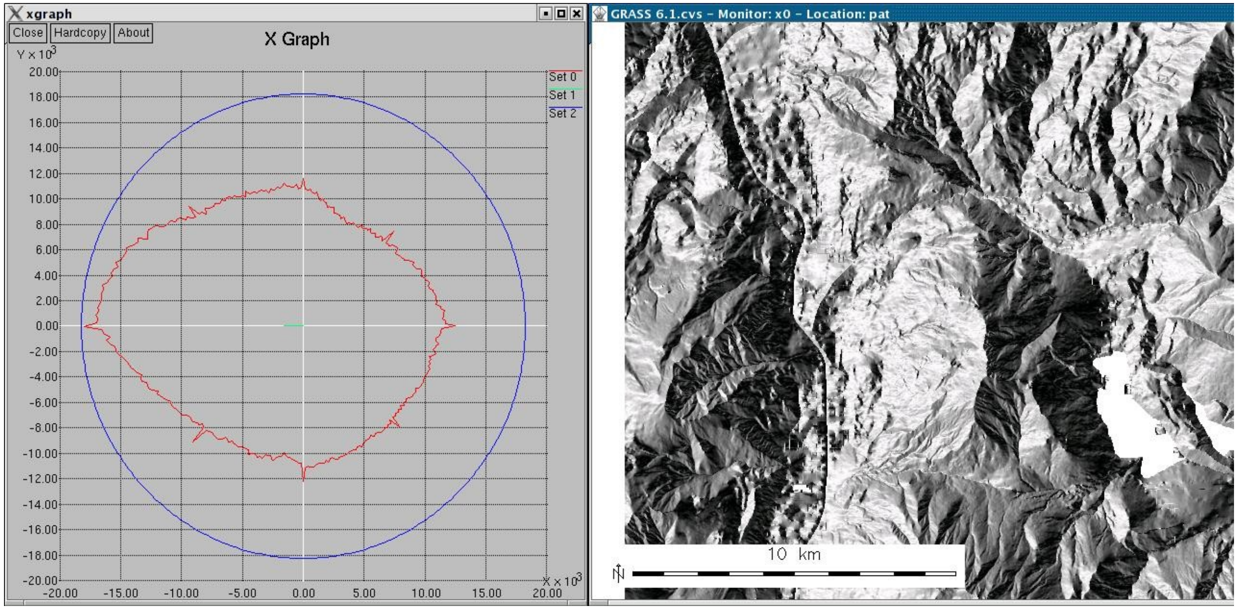
ITC-irst, Povo (Trento), Italy

(Document revised November 2006)

GRASS: Geographic Resources Analysis Support System

Scope: Learn the use of Free Software GIS

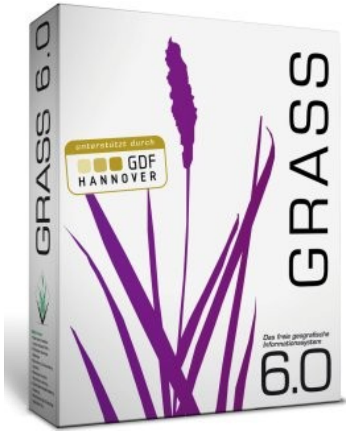
- PART I: Introduction to GRASS and QGIS
- PART II: Practical examples



GRASS: Geographic Resources Analysis Support System

Free Software GIS ("software libero"):

- GRASS master Web site is in Italy: **http://grass.itc.it**
- *Portable:* Versions for GNU/Linux, MS-Windows, Mac OSX, SUN, etc
- *Programming:* Programmer's Manual on Web site (PDF, HTML), generated weekly. Code is documented in source code files (doxygen)
- Sample data
- Mailing lists in various languages
- Commercial support

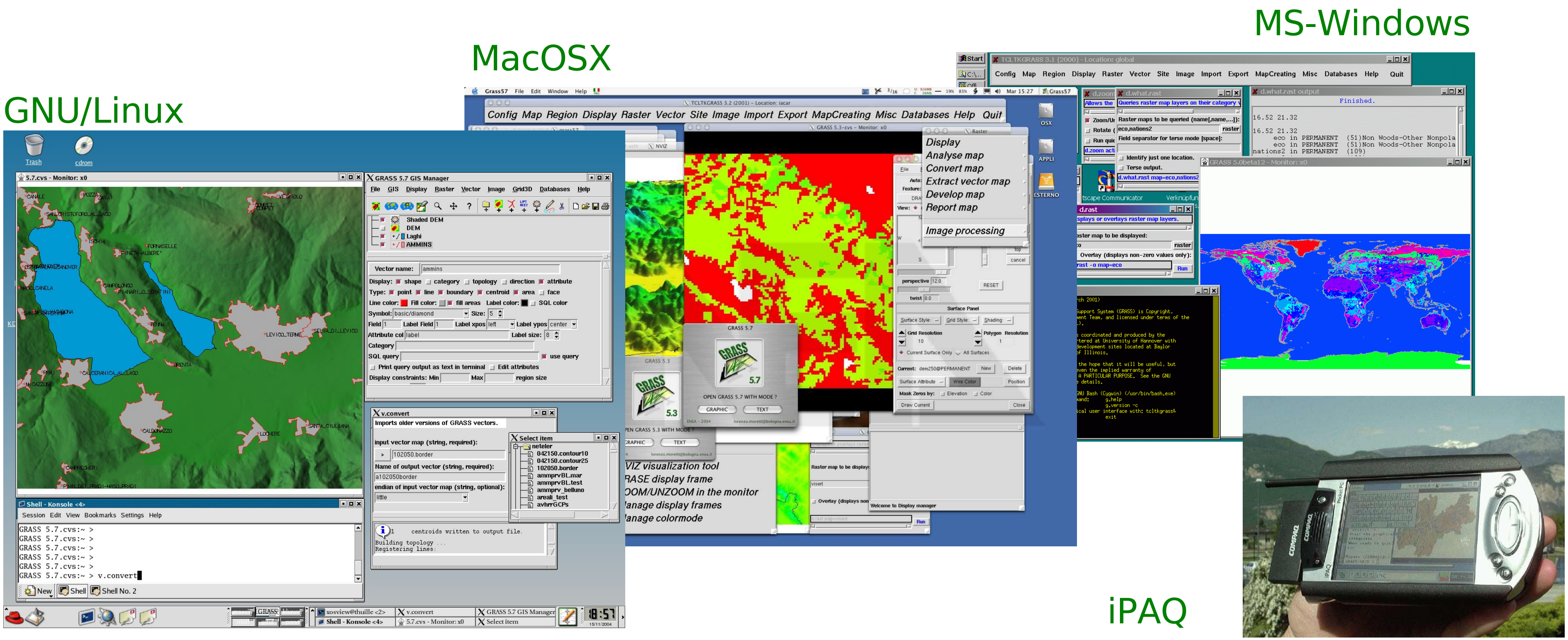


GRASS GIS

Brief Introduction

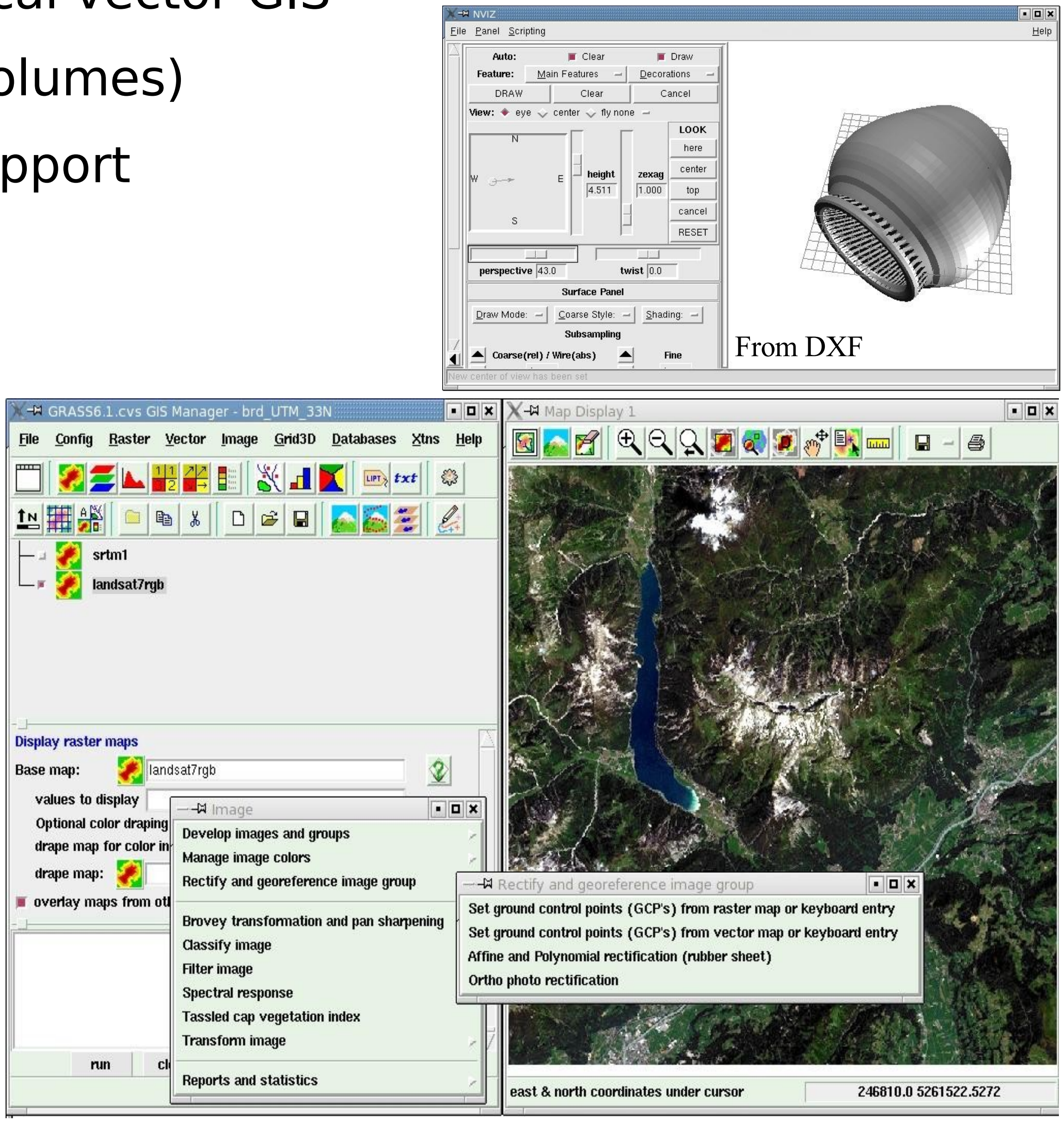
- Developed since 1984, **always Open Source**, since 1999 under GNU GPL
- Written in C programming language, **portable code** (multi-OS, 32/64bit)
- **International development team**, since 2001 coordinated at ITC-irst
- GRASS master Web site:

http://grass.itc.it



What's GRASS GIS?

- Raster and 2D/3D topological vector GIS
- Voxel support (raster 3D volumes)
- Vector network analysis support
- Image processing system
- Visualization system
- DBMS integrated (SQL) with dbf, PostgreSQL, MySQL and sqlite drivers
- In GRASS 6.1 translations of the user interface to 16 languages ongoing
- Interoperability: supports all relevant raster and vector formats



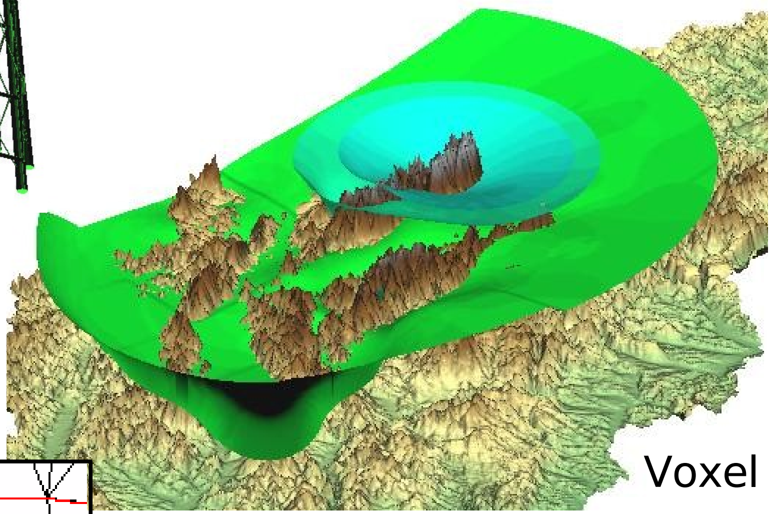
Spatial Data Types

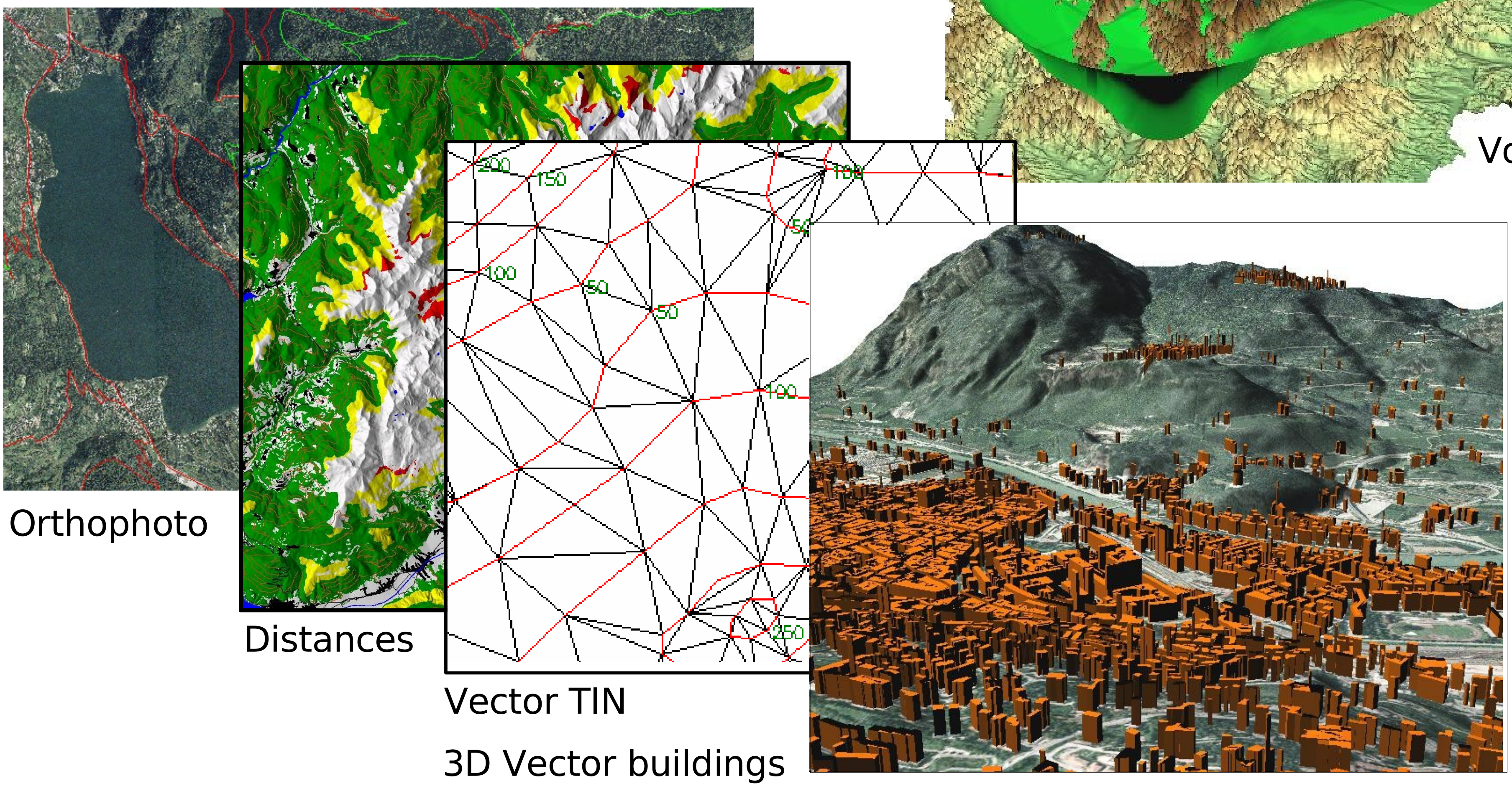
Supported Spatial Data Types

- 2D Raster data incl. image processing
- 3D Voxel data for volumetric data

- 2D/3D Vector data with topology
- Multidimensional points data

http://grass.itc.it





Raster data model

Raster geometry

- cell matrix with coordinates
- resolution: cell width / height   (can be in kilometers, meters, degree etc.)

1.0	2.0	2.0	3.0	3.0	4.0	4.0	5.0	5.0	5.0	6.0	6.0
0.0	1.0	1.0	2.0	2.0	3.0	3.0	4.0	4.0	4.0	5.0	5.0
0.0	0.0	0.0	1.0	1.0	2.0	2.0	3.0	3.0	3.0	4.0	4.0
1.0	0.0	0.0	0.0	0.0	1.0	1.0	2.0	2.0	2.0	3.0	3.0
2.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	2.0	2.0
3.0	2.0	2.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
4.0	3.0	3.0	2.0	2.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
5.0	4.0	4.0	3.0	3.0	2.0	2.0	1.0	1.0	1.0	0.0	0.0
6.0	5.0	5.0	4.0	4.0	3.0	3.0	2.0	2.0	2.0	1.0	1.0

y resolution

x resolution

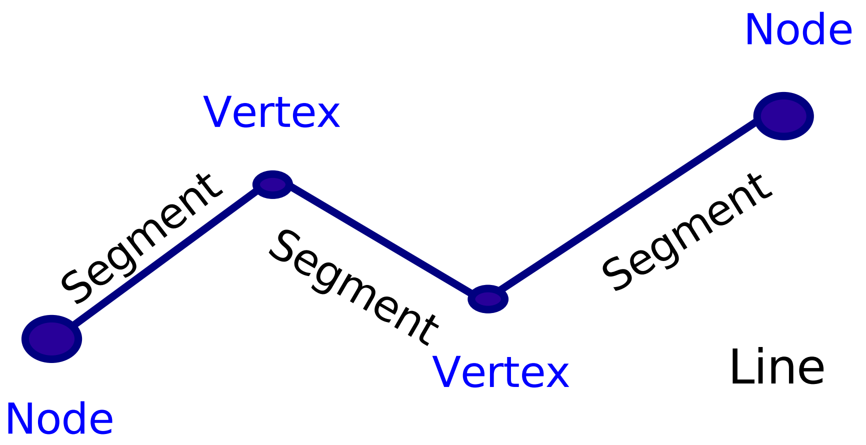
Vector data model

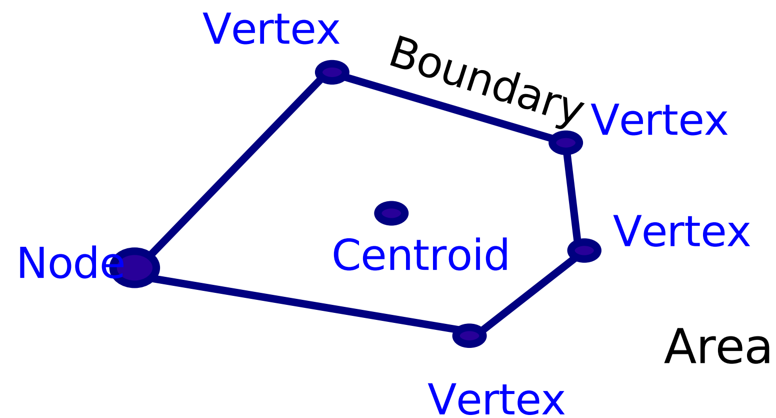
Vector geometry types

- Point
- Centroid
- Line
- Boundary
- Area (boundary + centroid)
- face (3D area)
- [kernel (3D centroid)]
- [volumes (faces + kernel)]

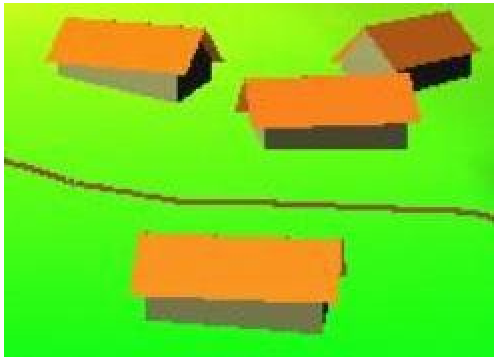
not in all GIS!

Geometry is **true** 3D: x, y, z





Faces



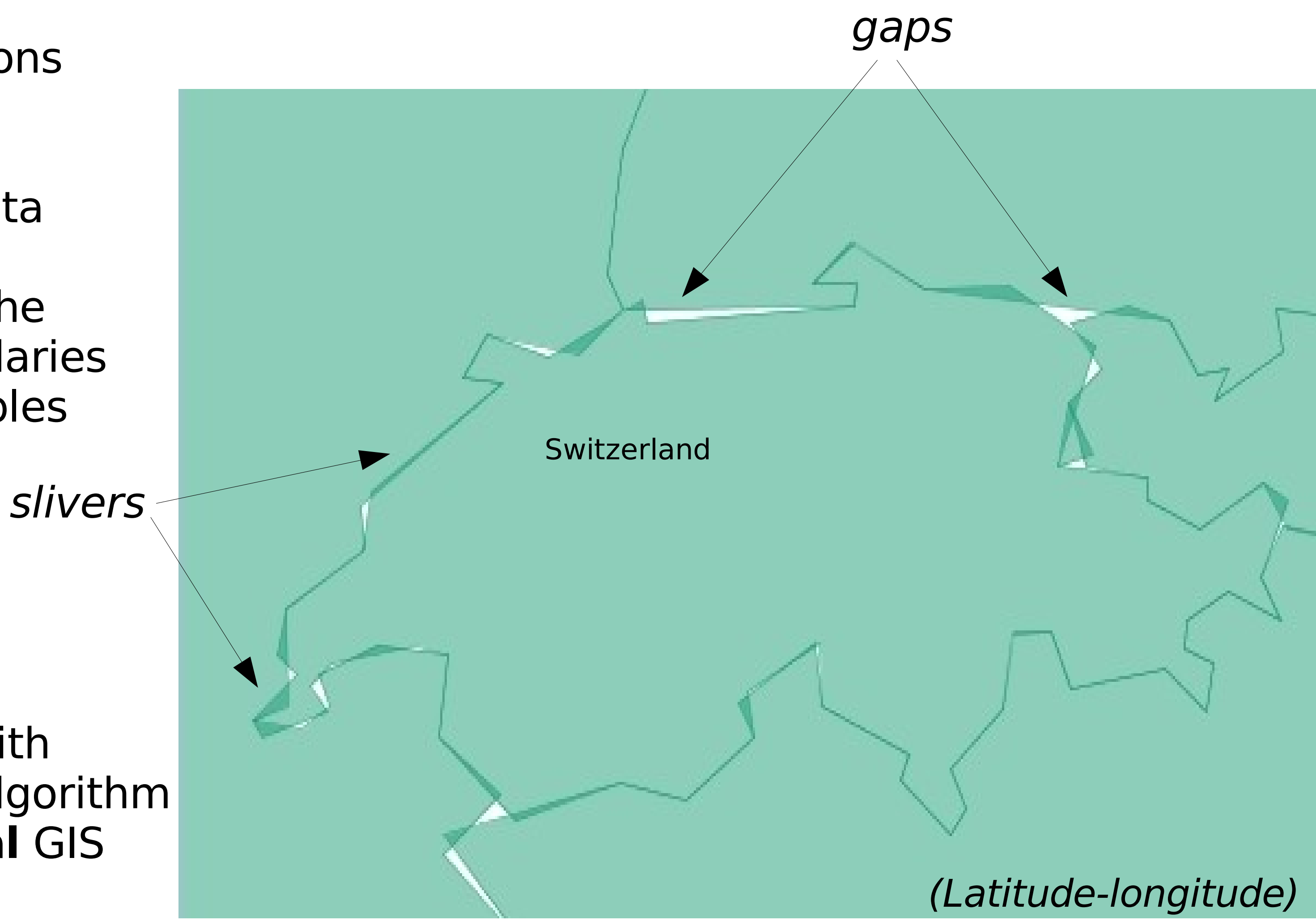
OGC Simple Features versus Vector Topology

Simple Features ...

- points, lines, polygons
- replicated boundaries for adjacent areas

Advantage:
- faster computations

Disadvantage:
- extra work for data maintenance
- in this example the duplicated boundaries are causing troubles



Map generalized with Douglas-Peucker algorithm in **non-topological** GIS

OGC Simple Features versus Vector Topology

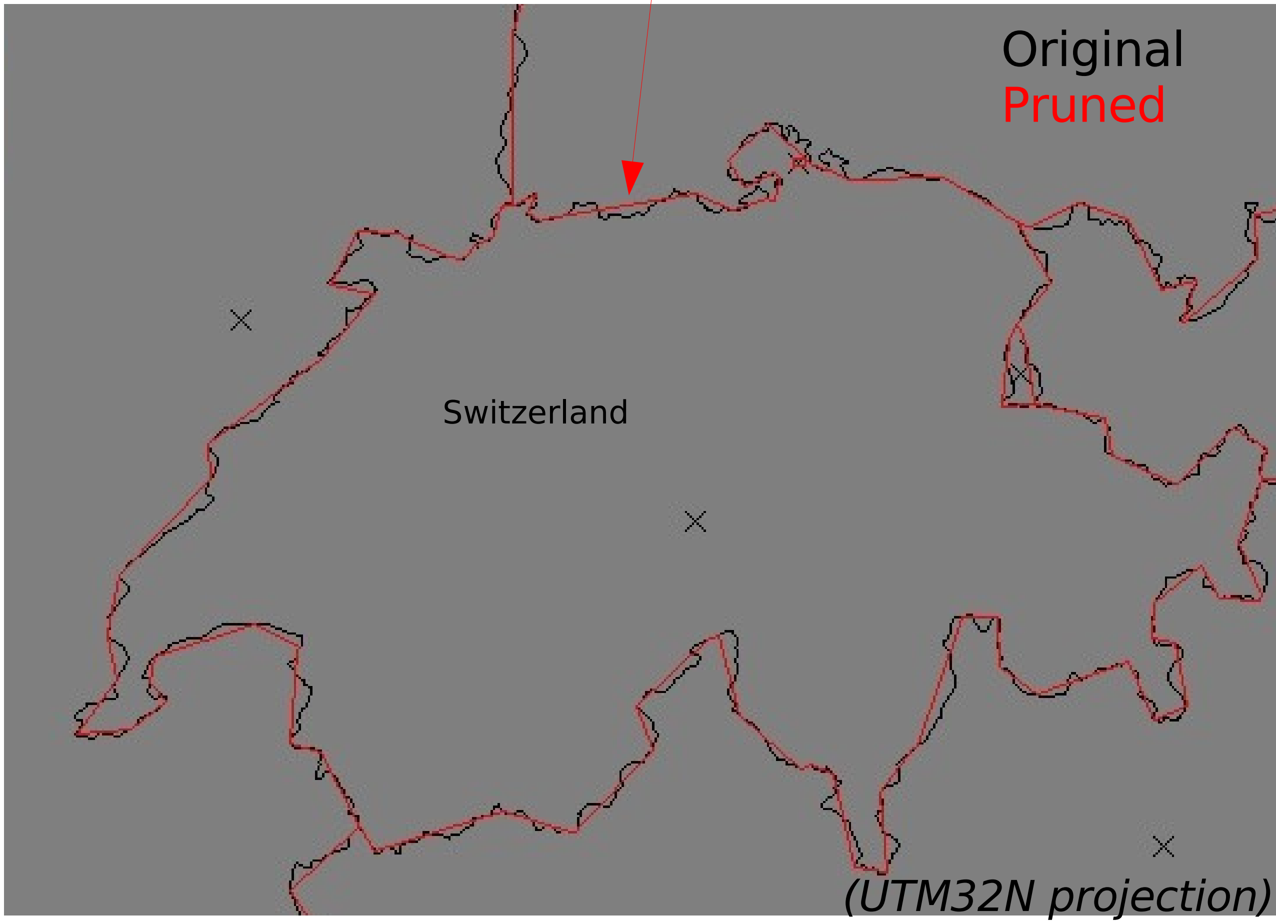
... versus Vector Topology

- points, centroids, lines, boundaries
- in topology centroid and boundary form an area
- single boundaries for adjacent areas

Advantage:
- less maintenance,
  high quality

Disadvantage:
- slower computations

each boundary is
a single line,
divided by two polygons



(UTM32N projection)

Map generalized with
v.clean "prune" algorithm
in **topological** GIS GRASS

Italy: Gauss-Boaga Coordinate System

Gauss-Boaga

- Transverse Mercator projection
- 2 zones (fuso Ovest, Est) with a width of 6º30' longitude
- Each zone is an own projection!

- False easting: Fuso Ovest: 1500000m (1500km)
  Fuso Est: 2520000m (2520km)
- False northing: 0m

- Scale along meridian: 0.9996 – secante case, not tangent case
- Ellipsoid: international (Hayford 1909, also called International 1924)

- Geodetic datum: Rome 1940 (3 national datums; local datums to buy from IGM). National datum values available at: `http://crs.bkg.bund.de/crs-eu/`

Italy: Gauss-Boaga Fuso Ovest

ESRI PRJ-File for Fuso Ovest (g.proj -w in GRASS)

```
PROJCS["Monte_Mario_Italy_1",
    GEOGCS["GCS_Monte_Mario",
        DATUM["Monte_Mario",
            SPHEROID["International_1924",6378388,297]],
        PRIMEM["Greenwich",0],
        UNIT["Degree",0.017453292519943295]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["False_Easting",1500000],
    PARAMETER["False_Northing",0],
    PARAMETER["Central_Meridian",9],
    PARAMETER["Scale_Factor",0.9996],
    PARAMETER["Latitude_Of_Origin",0],
    UNIT["Meter",1]]
```

EPSG codes:

Gauss-Boaga/Monte Mario 1: EPSG 26591
Gauss-Boaga/Monte Mario 2: EPSG 26592

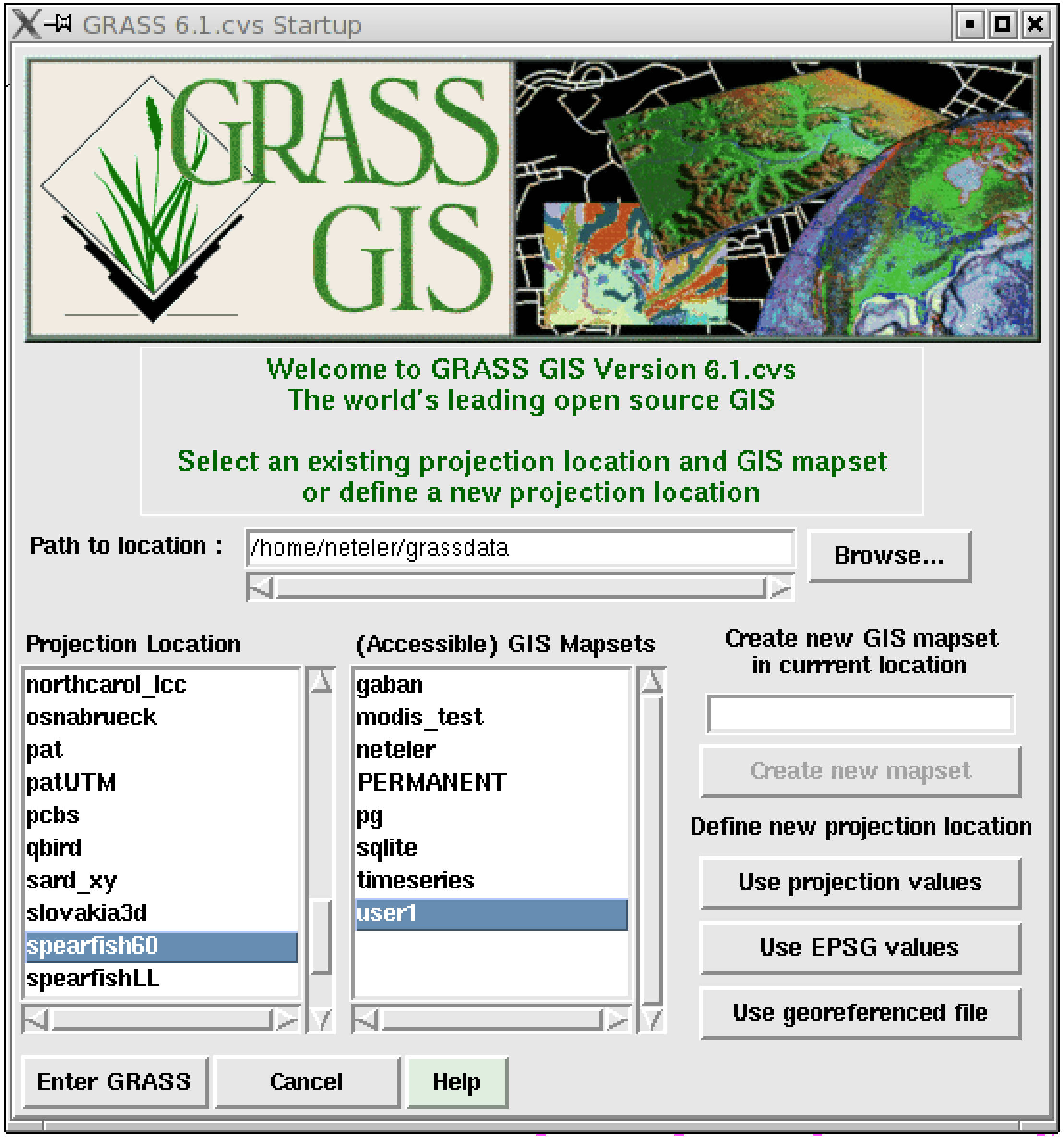
Geodetic Datums of Gauss-Boaga

Geodetic datum: “peninsular datum”

"Monte Mario to WGS 84 (4)","Position Vector 7-param. transformation",
"X-axis translation","1","-104.1","metre","Italy - mainland"
"Y-axis translation","2","-49.1","metre","Italy - mainland"
"Z-axis translation","3","-9.9","metre","Italy - mainland"
"X-axis rotation","4","0.971","arc-second","Italy - mainland"
"Y-axis rotation","5","-2.917","arc-second","Italy - mainland"
"Z-axis rotation","6","0.714","arc-second","Italy - mainland"
"Scale difference","7","-11.68","parts per million","Italy - mainland"

also available: Sardegna, Sicilia

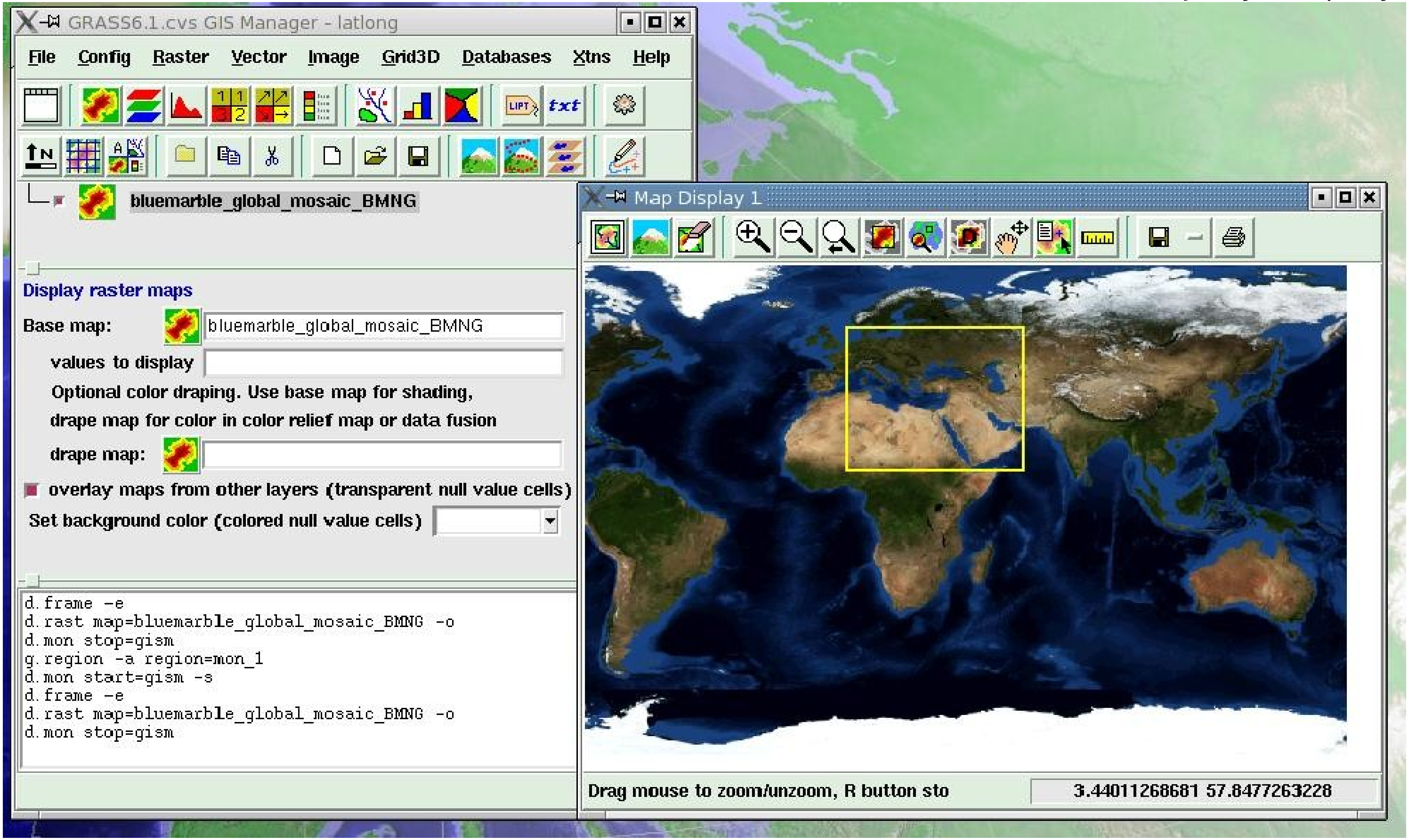
How to use GRASS GIS?



GRASS startup screen

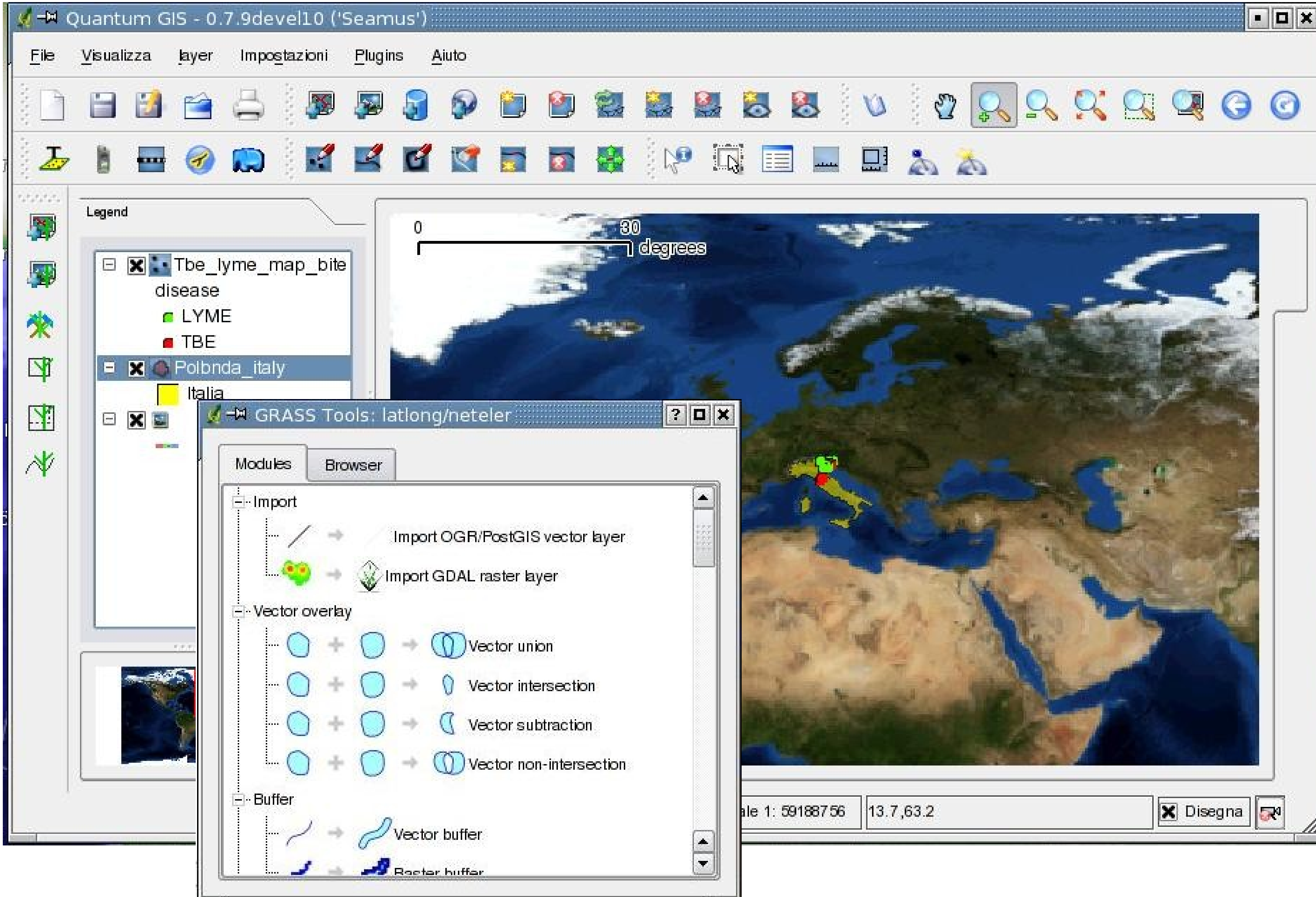
GRASS: Modernized GIS manager and WMS support

gis.m: Michael Barton, Cedric Shock
r.in.wms: Sören Gebbert & Jachym Cepicky



GRASS integration with QGIS

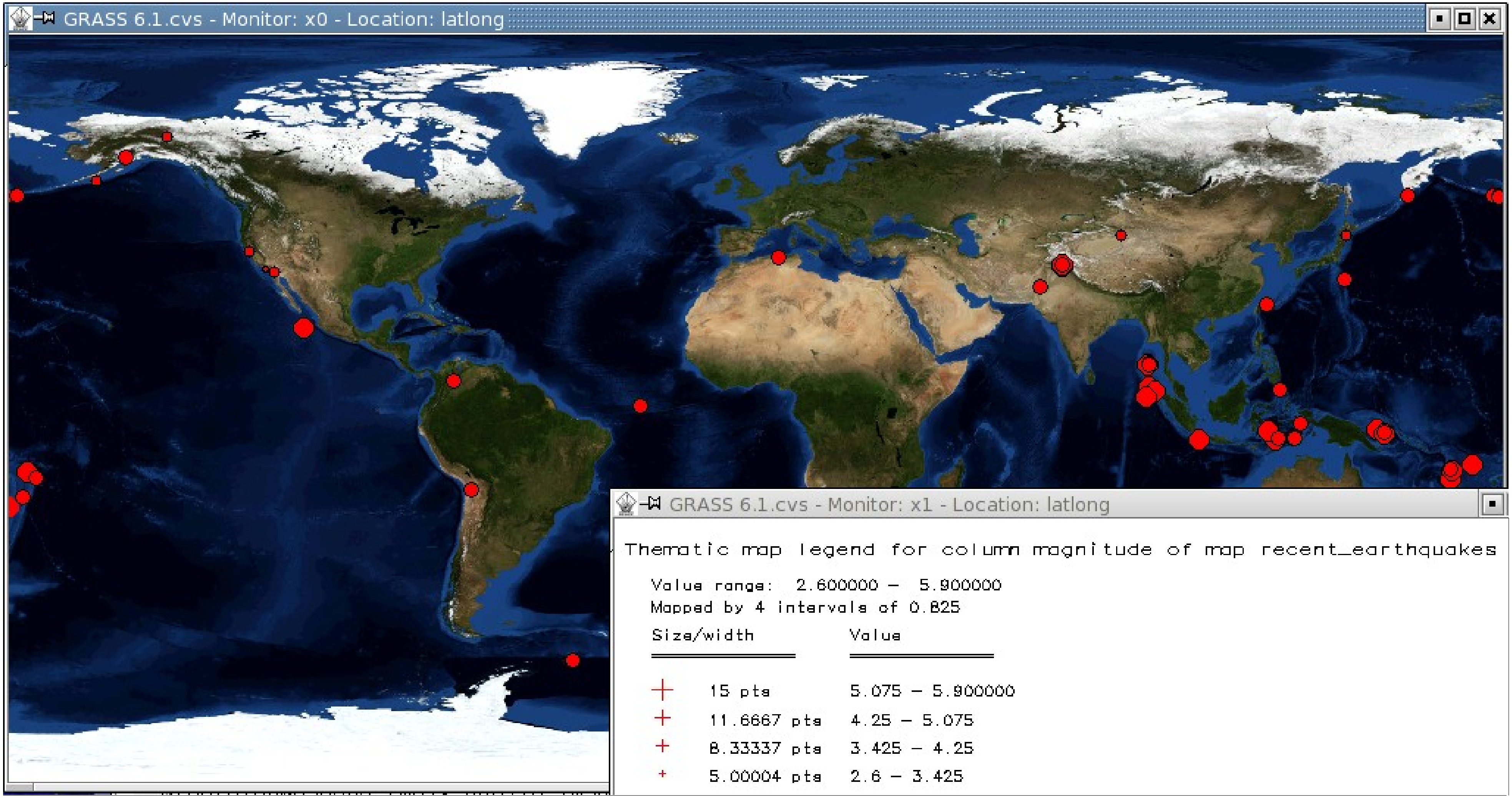
QGIS-GRASS plugin: Radim Blazek http://qgis.org



WebGIS: Integration of data sources

GRASS in the Web

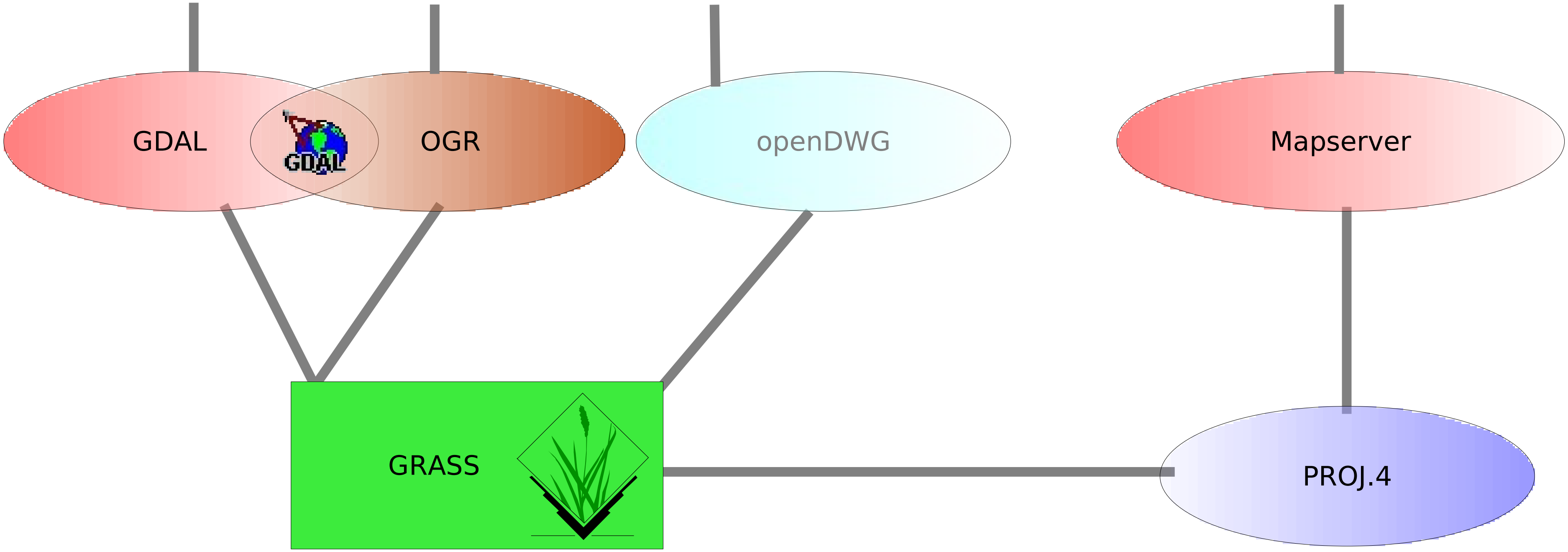
Real-time monitoring of Earthquakes (provided in Web by USGS)
with GRASS/PHP: http://grass.itc.it/spearfish/php_grass_earthquakes.php



GRASS GIS Interoperability

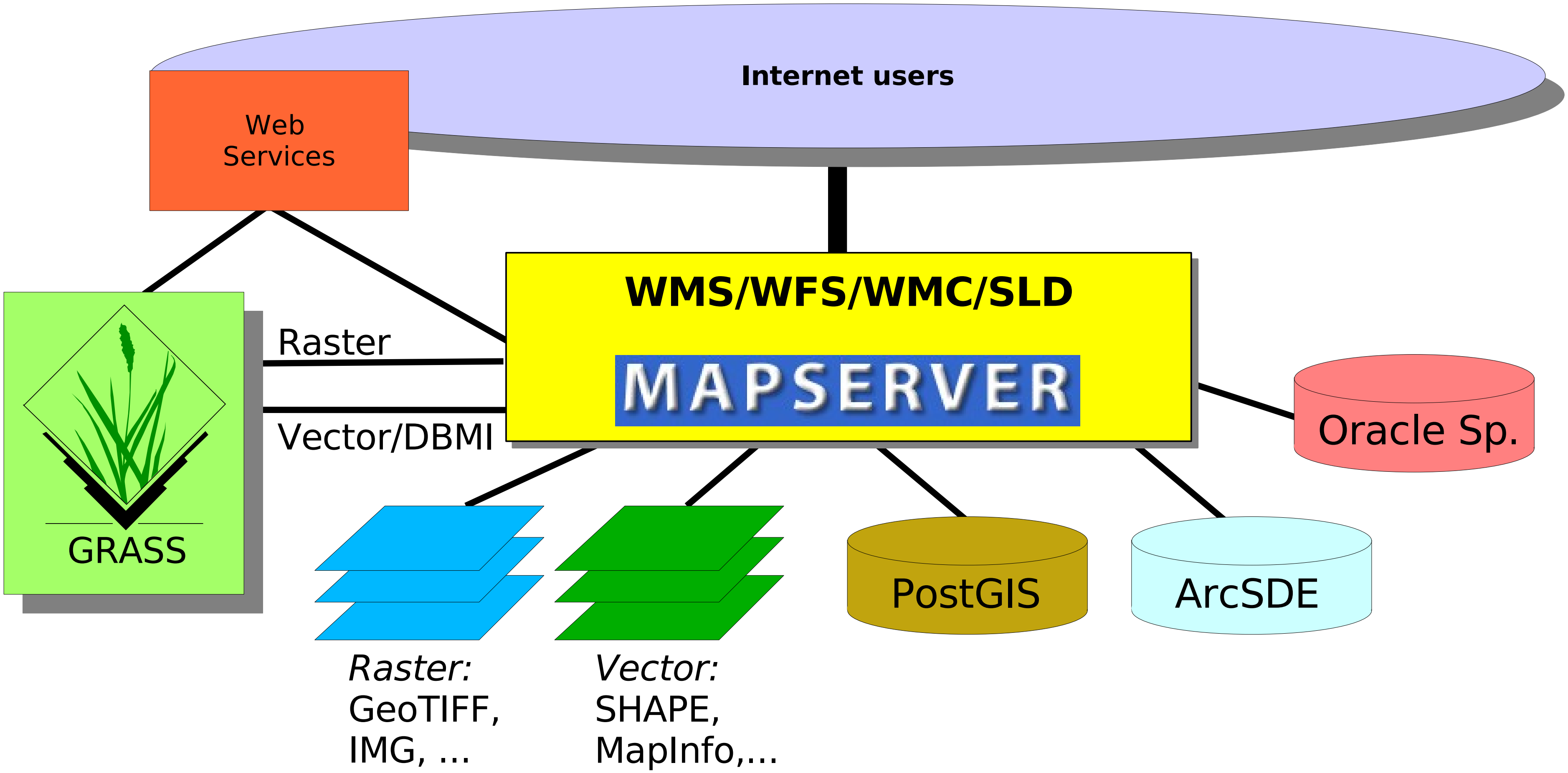
Data models and formats

Raster	Vector	CAD	WebGIS
GeoTIFF	DGN	DXF	Web Map Service (WMS)
Erdas IMG	ESRI-SHAPE	DWG	Web Coverage Service (WCS)
MrSID	GML	...	Web Feature Service (WFS)
ECW	Spatial SQL		Web Map Context Documents (WMC)
JPEG2000	...		
...			

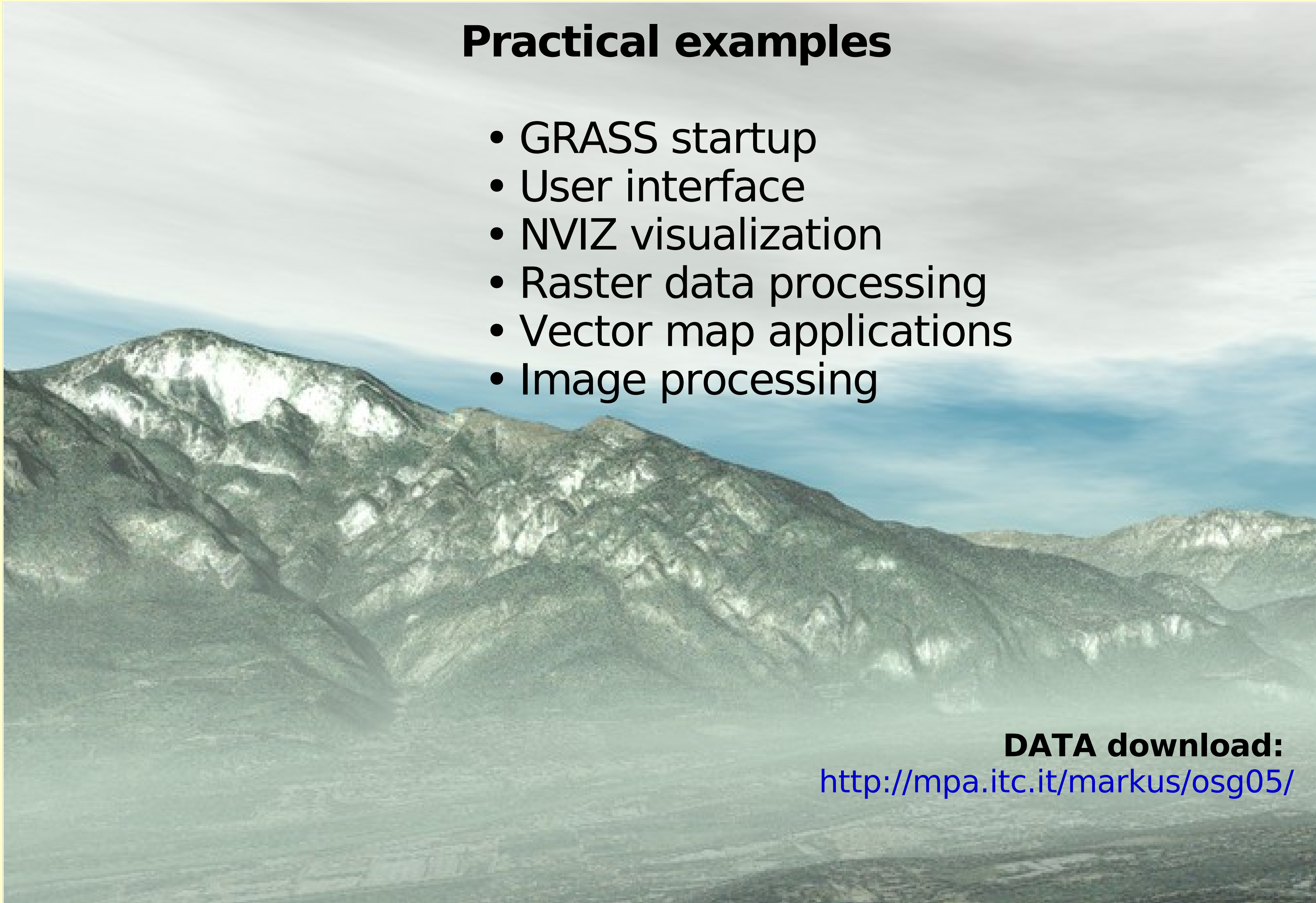


WebGIS: Integration of data sources

GIS – DBMI – Mapserver linking



Part II



Practical examples

- GRASS startup
- User interface
- NVIZ visualization
- Raster data processing
- Vector map applications
- Image processing

DATA download:
http://mpa.itc.it/markus/osg05/

Command structure

GRASS Command Overview			
prefix	*function class*	*type of command*	*example*
d.*	display	graphical output	*d.rast: views raster map* *d.vect: views vector map*
db.*	database	database management	*db.select: selects value(s) from table*
g.*	general	general file operations	*g.rename: renames map*
i.*	imagery	image processing	*i.smap: image classifier*
ps.*	postscript	map creation in Postscript format	*ps.map: map creation*
r.*	raster	raster data processing	*r.buffer: buffer around raster features* *r.mapcalc: map algebra*
r3.*	voxel	raster voxel data processing	*r3.mapcalc: volume map algebra*
v.*	vector	vector data processing	*v.overlay: vector map intersections*

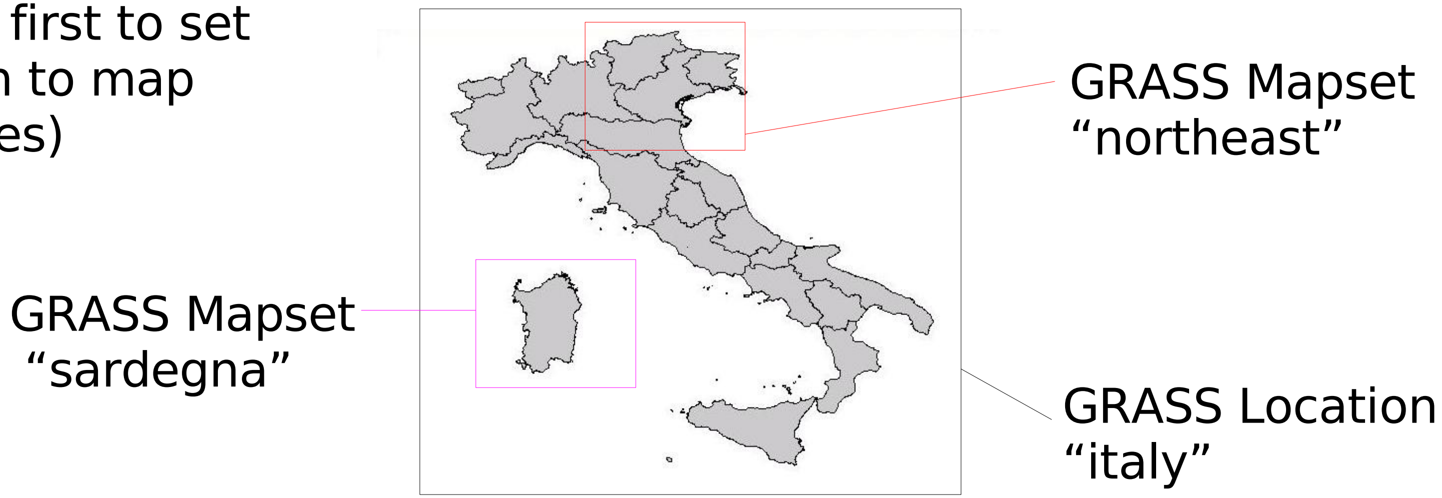
Some things you should know about GRASS

- Import of data: GRASS always import the **complete** map

- Export of data:
  - Vector maps: always the entire map is exported (cut before if needed)
  - Raster maps: r.out.gdal always exports entire map at original resolution
    r.out.tiff (etc.) export at current region and resolution

What's a region in GRASS?

- The **default** region is the standard settings of a GRASS location which is essentially independent from any map
- A region is the current working area (user selected resolution and coordinate boundaries)
- All **vector** calculations are done at full vector map
- All **raster** calculations are done at current resolution/region. To do calculations at original raster map resolution/region, the easiest way is to use 'g.region' first to set current region to map (see next slides)



Spearfish Sample Dataset

Spearfish (SD) sample data location

Maps:

- raster, vector and point data
- covering two 1:24000 topographic maps (quadrangles Spearfish and Deadwood North)
- UTM zone 13N, transverse mercator projection, Clarke66 ellipsoid,
- NAD27 datum, metric units, boundary coordinates: 4928000N, 4914000S, 590000W, 609000E

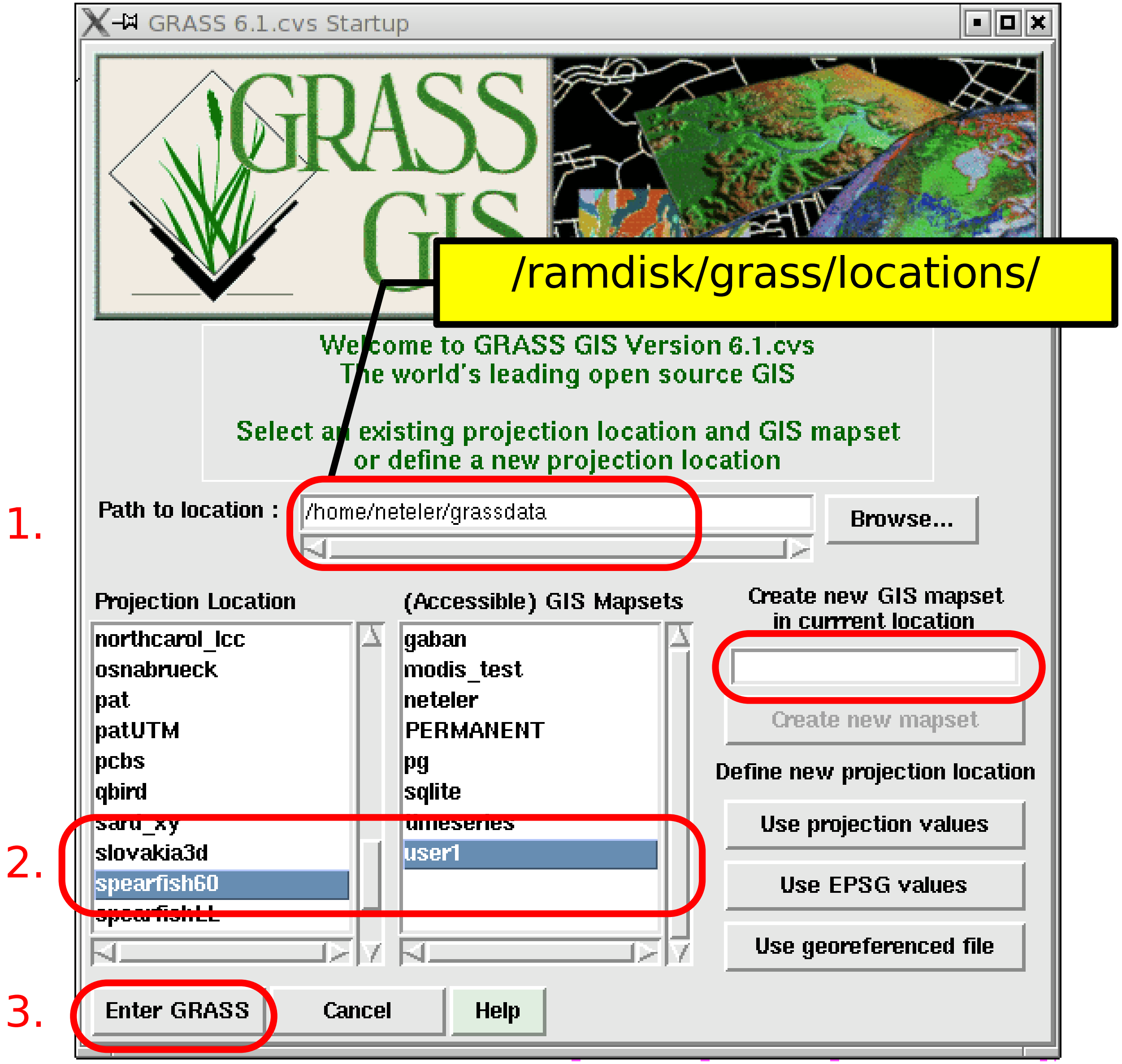
DATA download:
http://mpa.itc.it/markus/osg05/

Practical GIS Usage

- Start a "terminal"
  to enter commands

- Start GRASS 6 within
  the terminal:

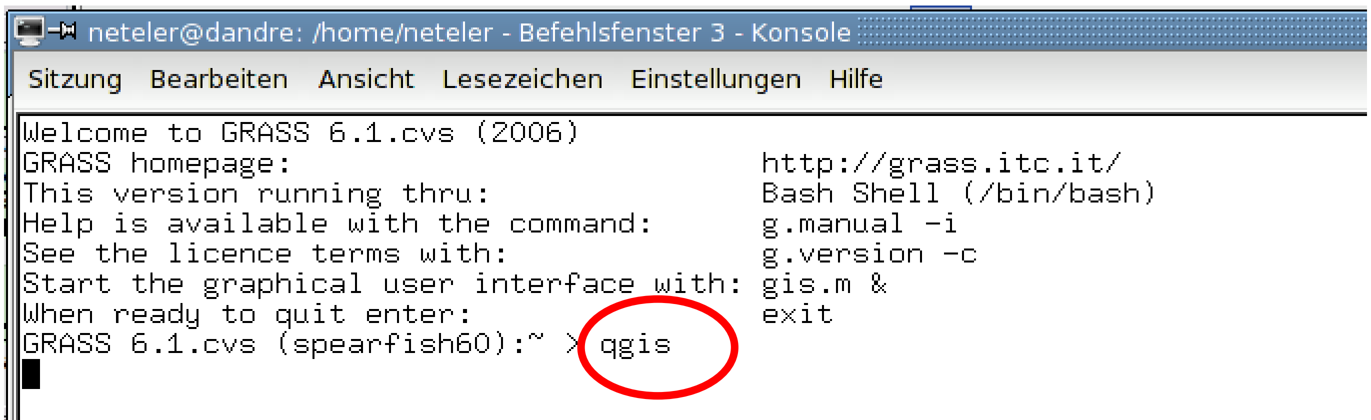
  ```
  grass61 -help

  grass61 -gui
  ```

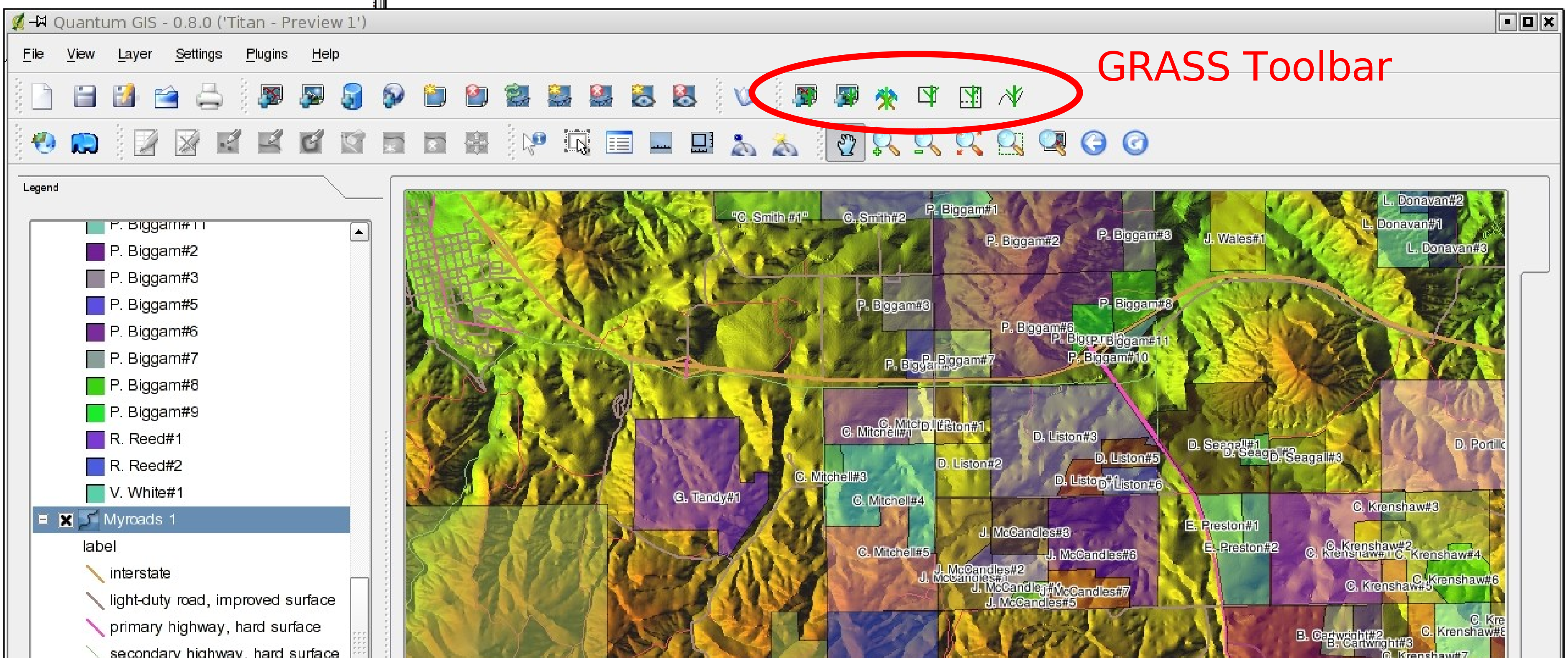


GRASS user interface: QGIS

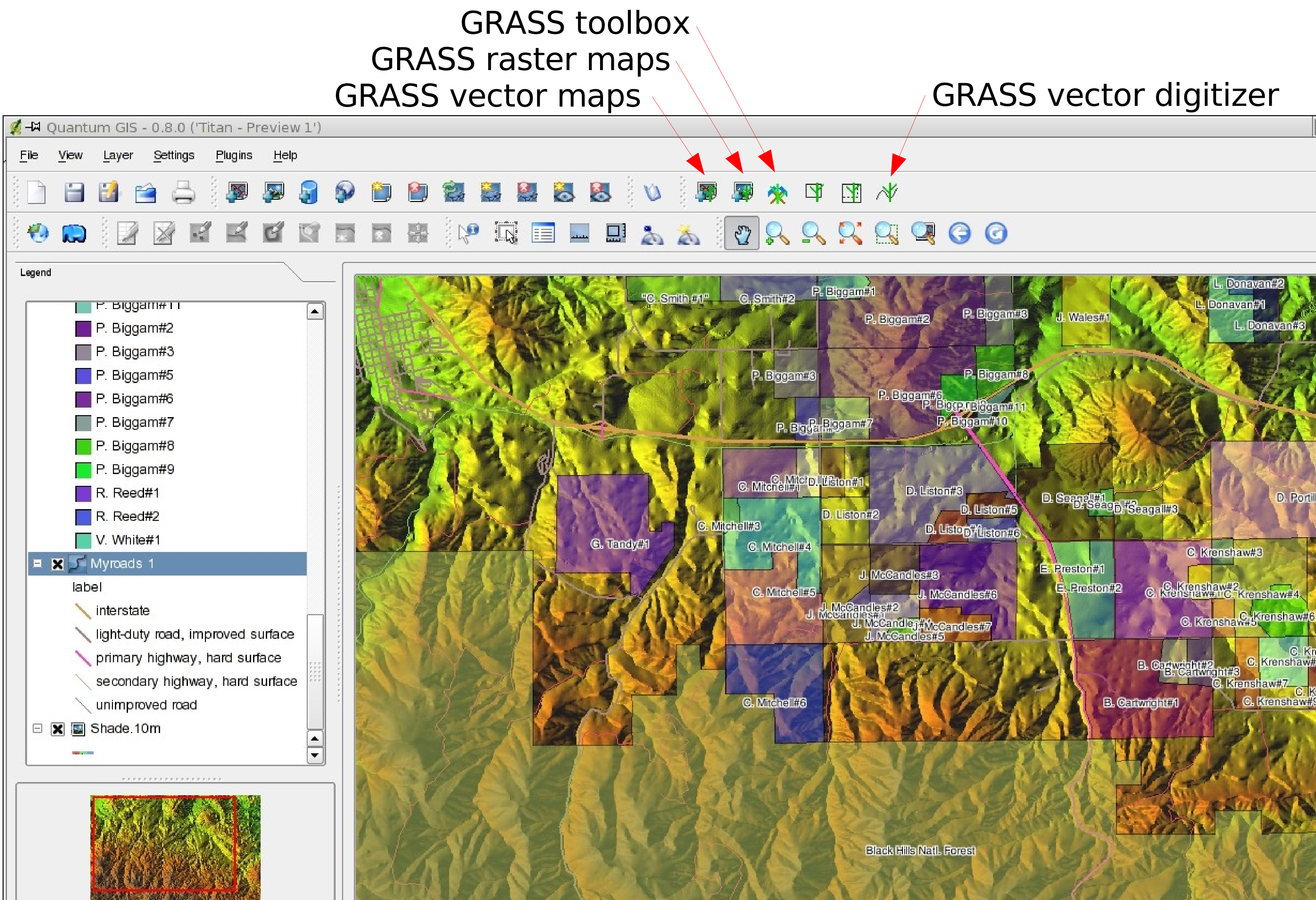
http://qgis.org

- Start QGIS within GRASS terminal:
  `qgis`





QGIS: further key functionality



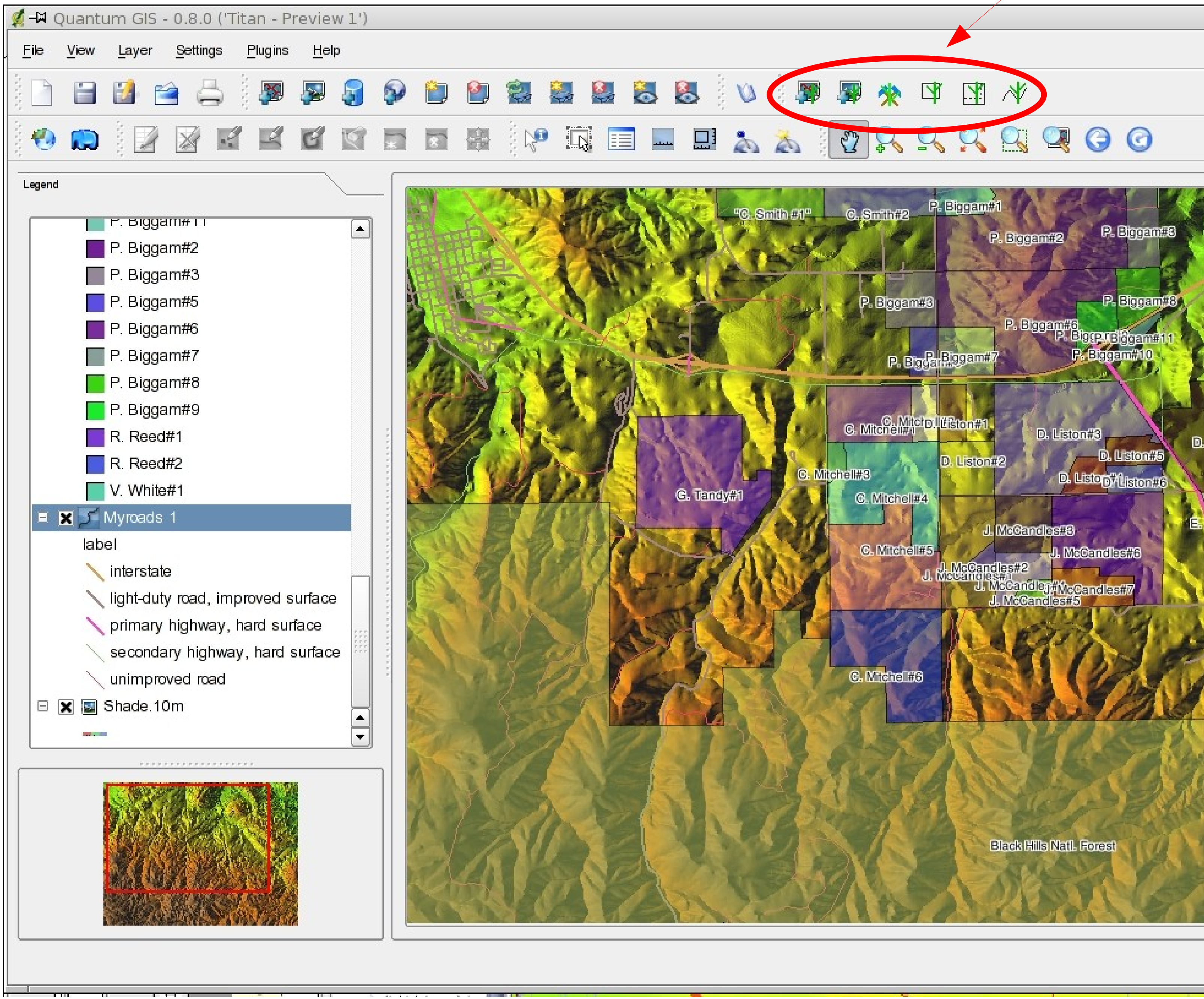
New GRASS user interface: QGIS

Excercise:

Please reproduce this map view!

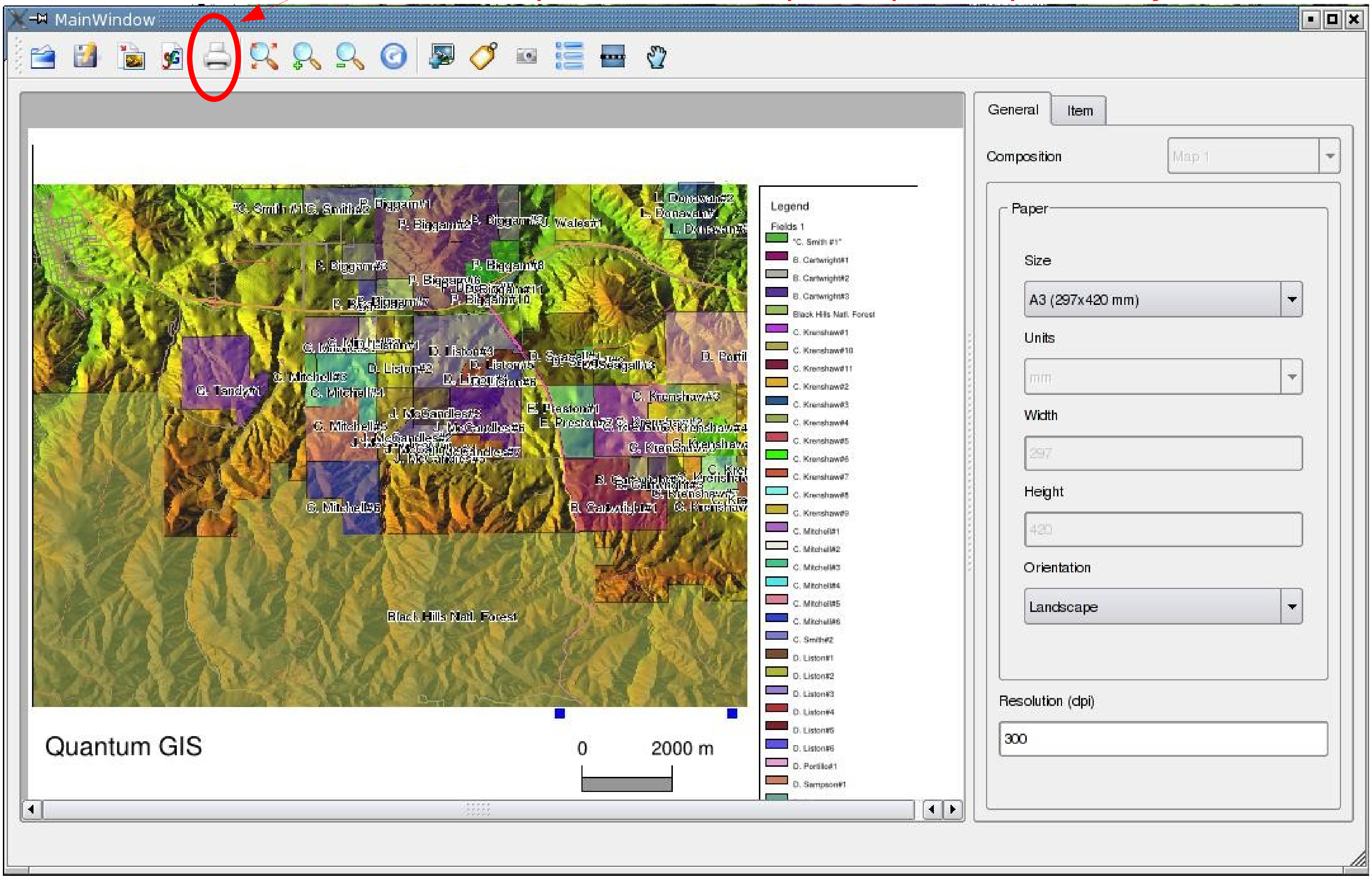
Raster:
- elevation.dem
- aspect

Vector:
- roads
- fields



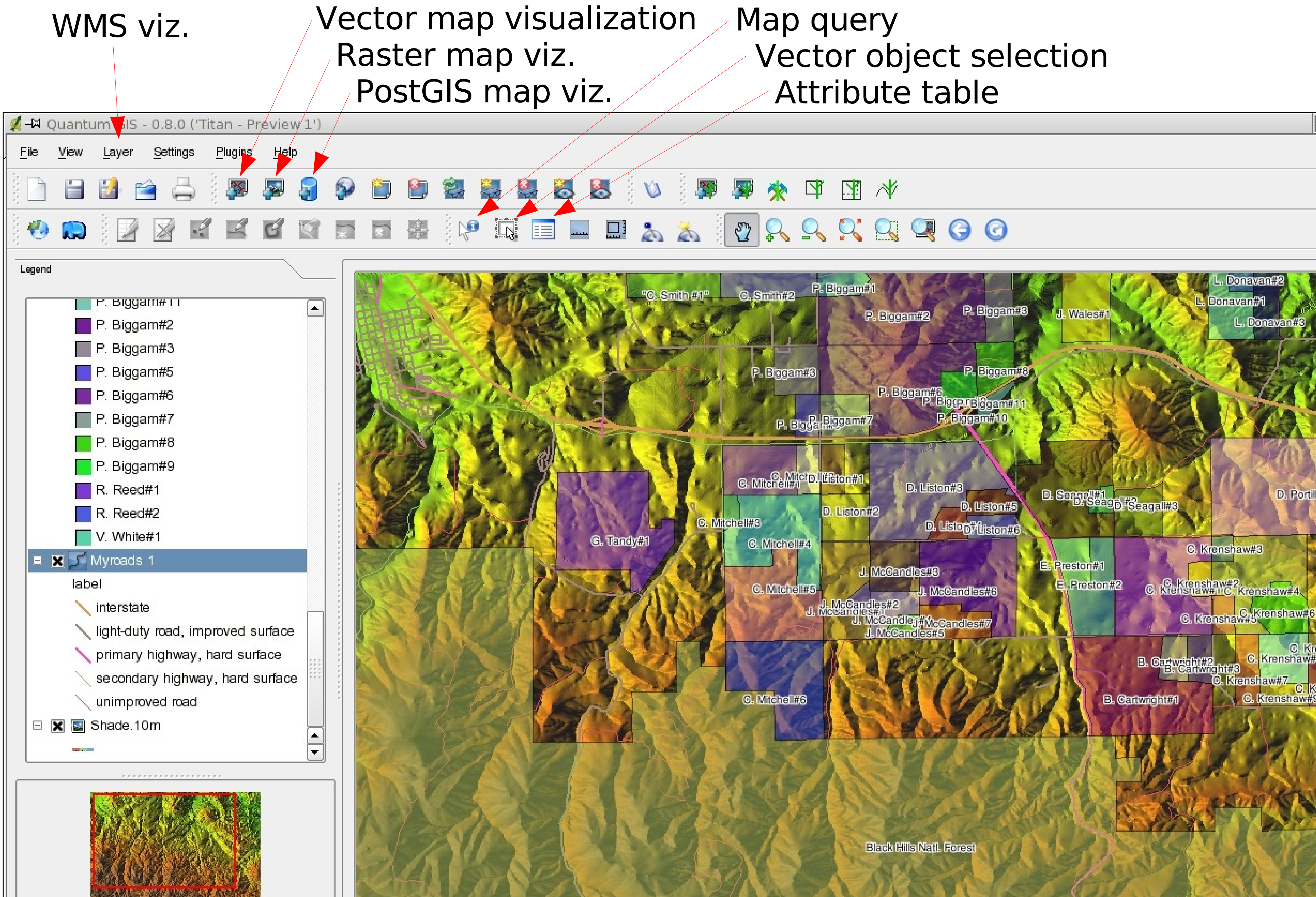
QGIS map composer: prepare map with layout

Transfer map view into map composer (printer symbol)



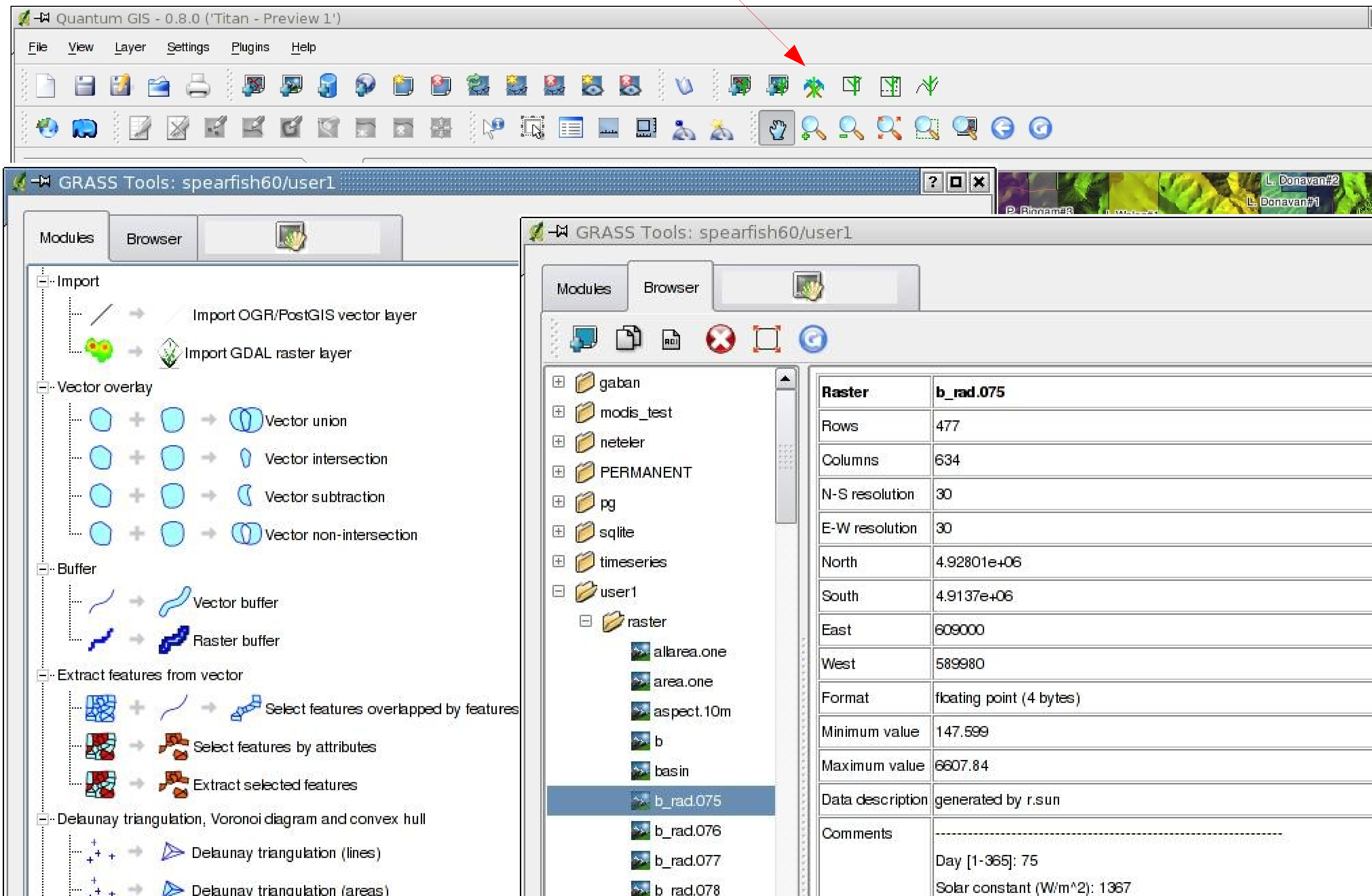
Creating a paper map for printing or saving into a file (SVG, PNG, Postscript)

QGIS: further key functionality



QGIS: GRASS toolbox

GRASS toolbox



QGIS-GRASS Exercises: Noise impact 1/4

1) Simple noise impact map:

Extract interstate (highway) from *roads* vector map into new map
and buffer interstate for 3km in each direction

GRASS commands:

a) first look at the table to get column name and ID of interstate:
v.db.select roads

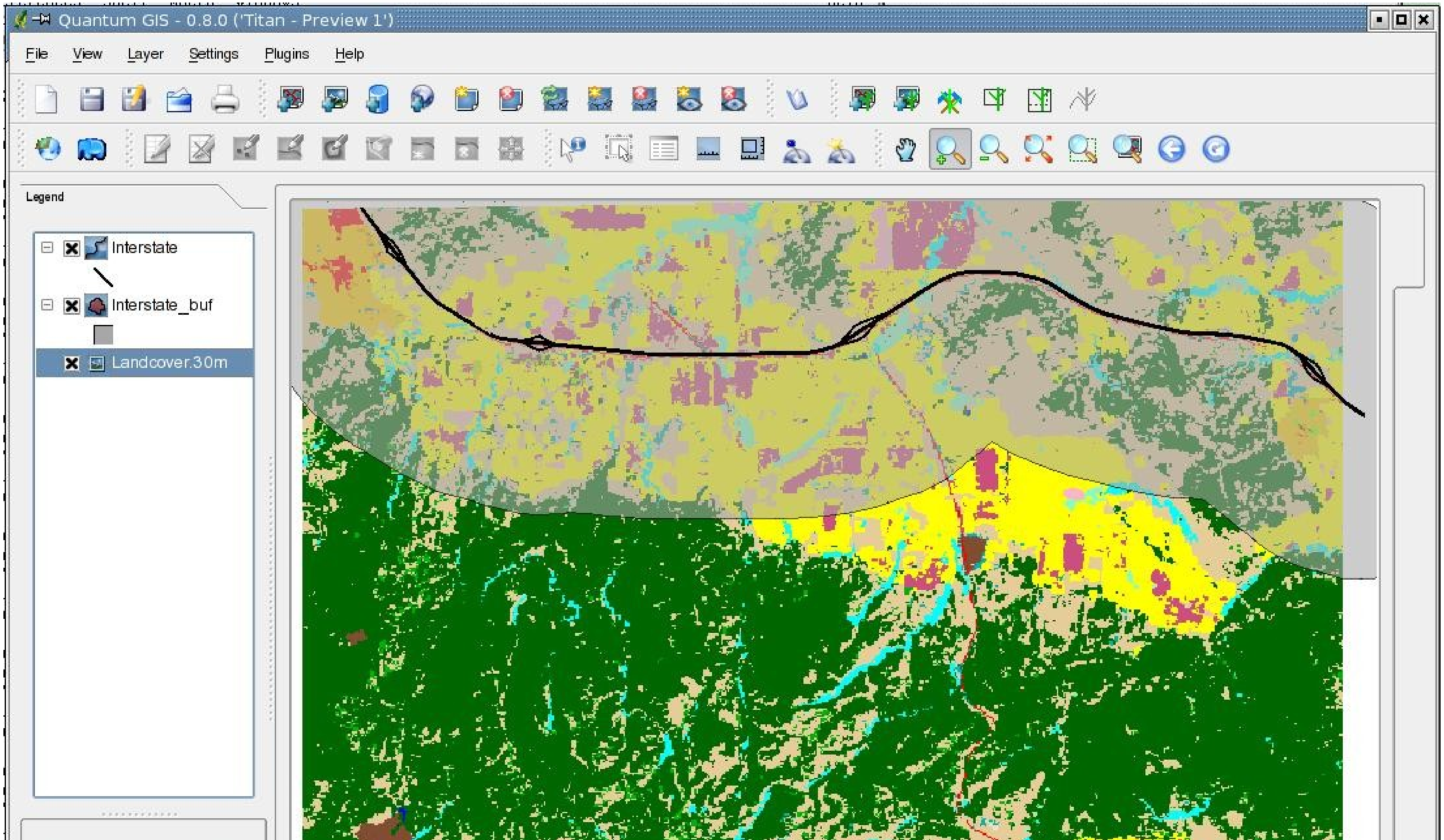
b) we extract only 'interstate' (cat = 1, cat is the GRASS standard column name for ID):
v.extract in=roads out=interstate where="cat = 1"

c) we buffer the interstate (give buffer in map units which is meters here):
v.buffer interstate out=interstate_buf3000 buffer=3000

QGIS-GRASS Exercises: Noise impact 2/4

2) Verify affected areas:

Look at *landcover.30m* raster map,
overlay extracted *interstate*
and overlay *buffered interstate_buf3000* (use transparency to make it nice)

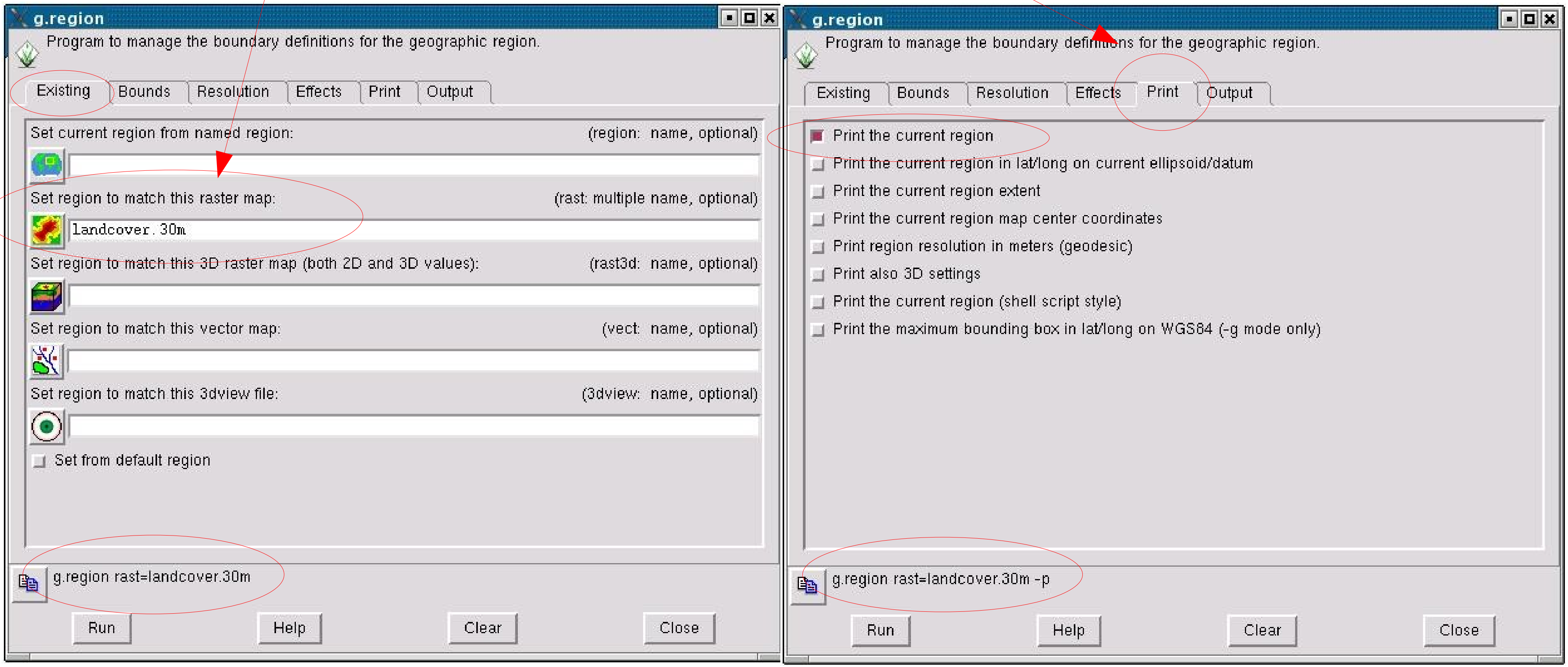


Info: Command line versus graphical user interface

On the next slide we either use the following command line:

```
# set current region to landcover map, '-p' prints the settings:
g.region rast=landcover.30m -p
```

or these settings in the graphical user interface:



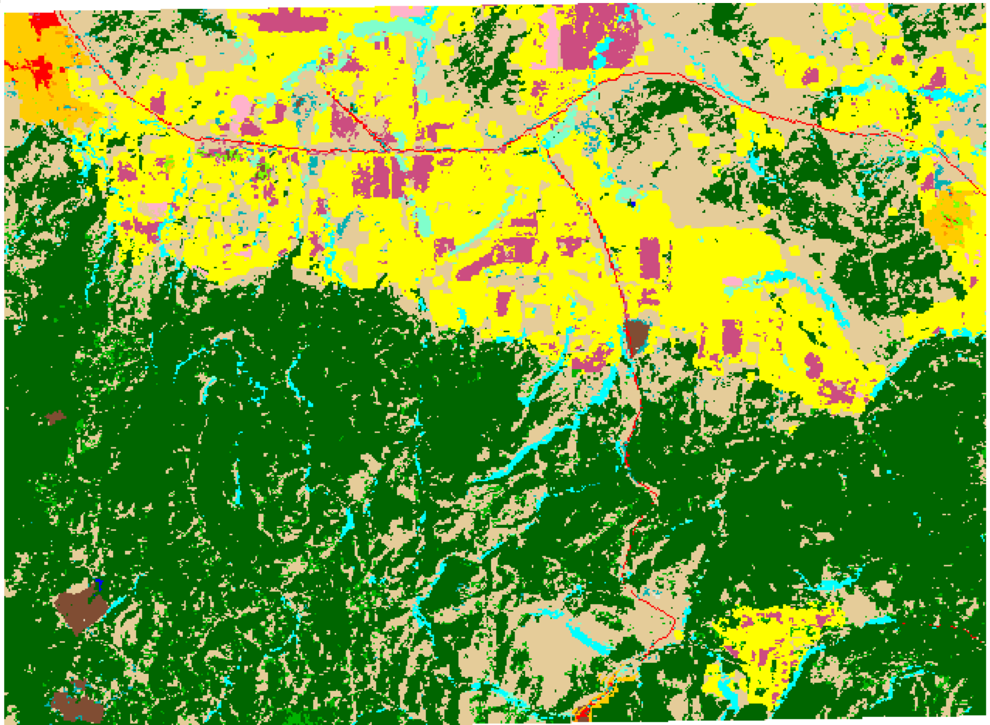
QGIS-GRASS Exercises: Noise impact 3/4

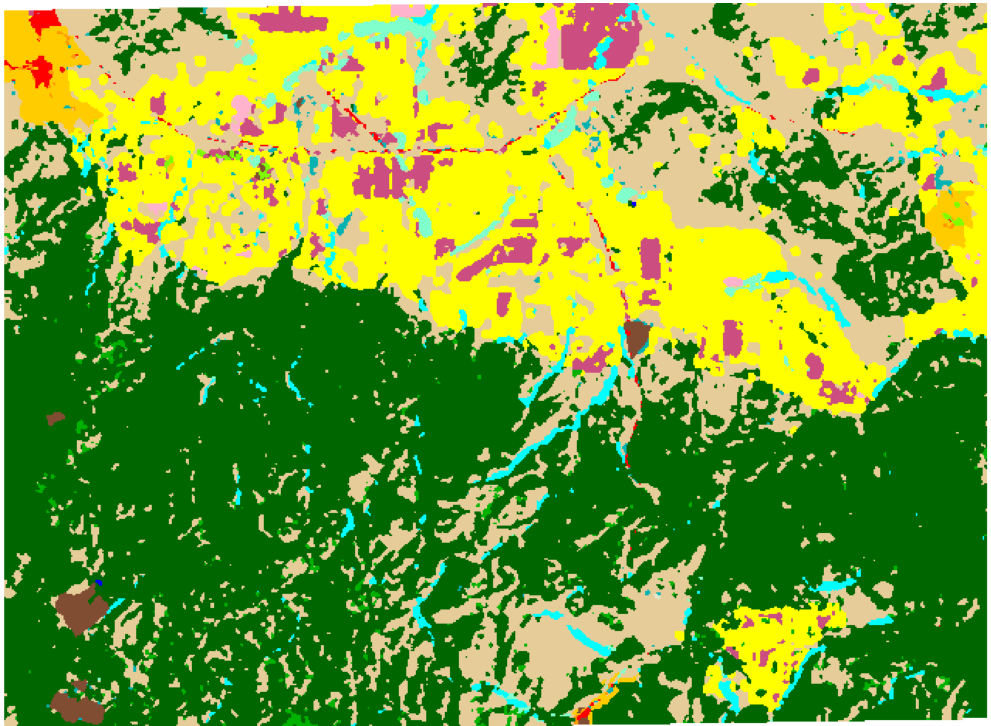
How to get statistics on influenced landcover-landuse units?
-> needs generalization of original *landcover.30m* map (originates from satellite map)

Approach 1:
Raster based generalization: "mode" operator in moving window

```
# set current region to landcover map, '-p' prints the settings:
g.region rast=landcover.30m -p
r.neighbors in=landcover.30m out=landcover.smooth method=mode size=3
```

3x3 moving window



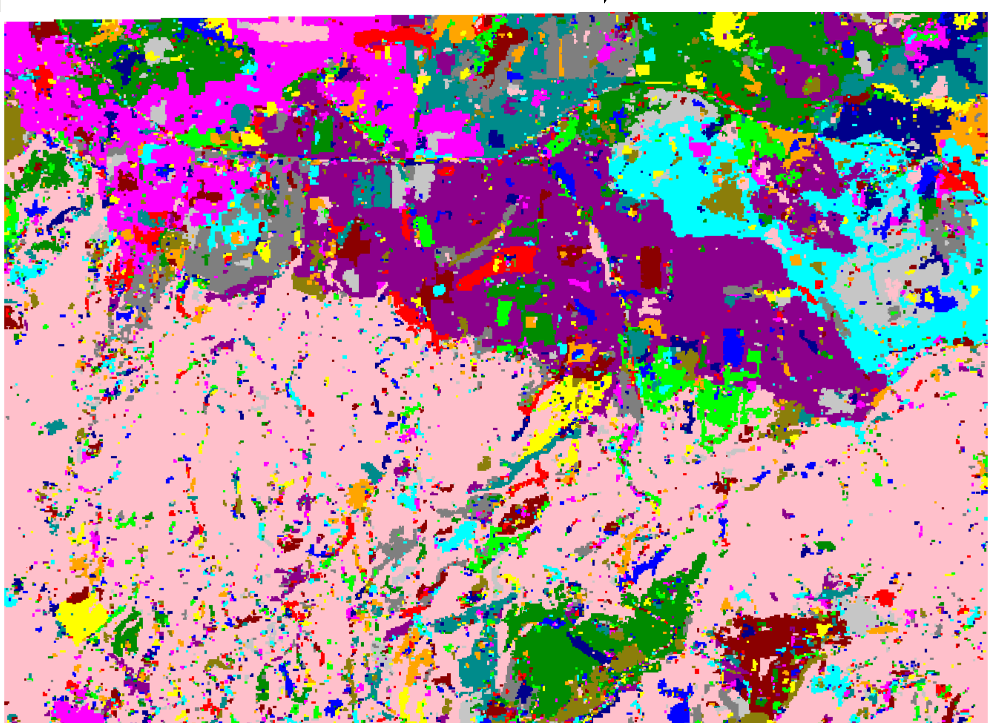


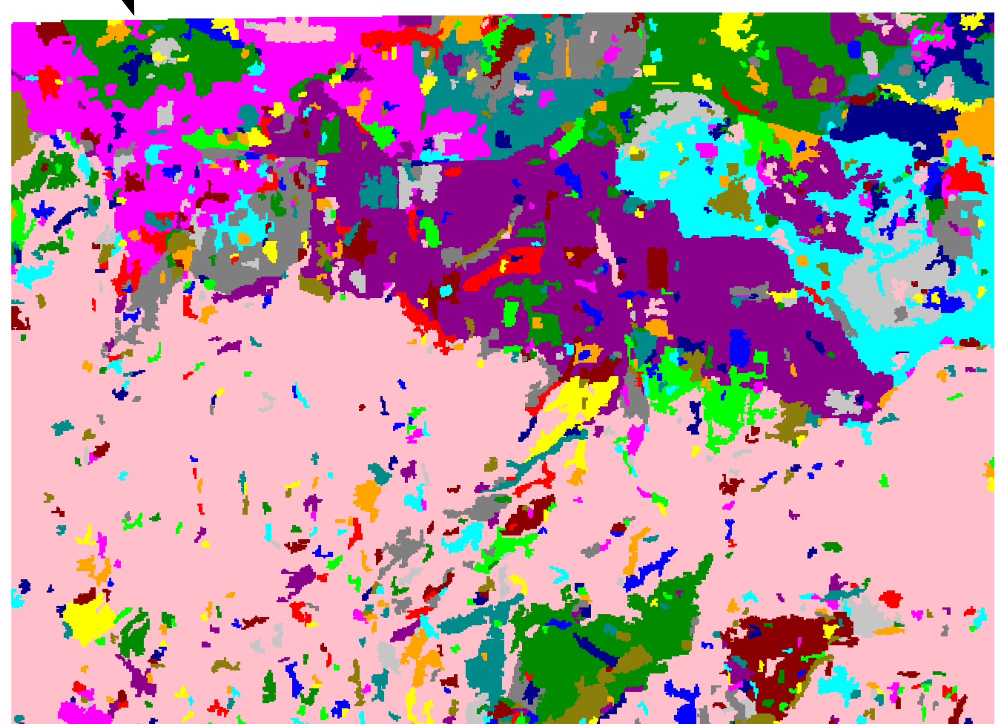
QGIS-GRASS Exercises: Noise impact 4/4

... Generalization cont'ed:

Approach 2:
Vector based generalization: “rmarea” tool: merges small areas into bigger a.

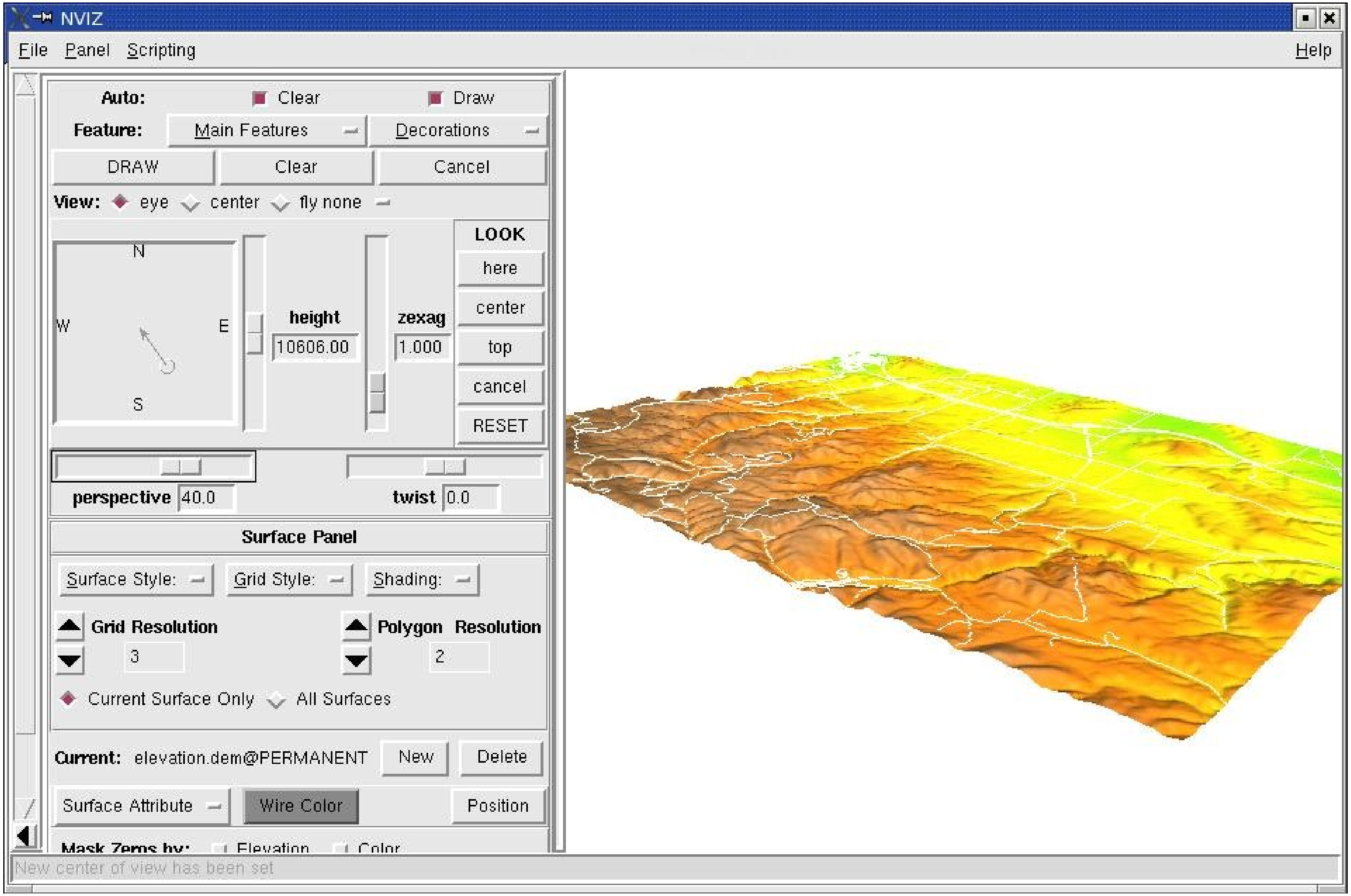
```
# zoom to map:
g.region rast=landcover.30m  -p
# raster to vector conversion:
r.to.vect in=landcover.30m out=landcover_30m f=area
# filter perimeter of 3x3 pixels ( threshold=(30 * 3)^2 = 8100)
v.clean in=landcover_30m out=landcover_30m_gen tool=rmarea thresh=8100
```





Perspective view of maps

- `nviz el=elevation.dem vect=roads`



Location and Mapset: “GRASS speech”

Database: contains all GRASS data

Each GRASS project is organized in a „Location“ directory with subsequent „Mapset(s)“ subdirectories:

Location: contains all spatial/attribute data of a geographically defined region (= project area)

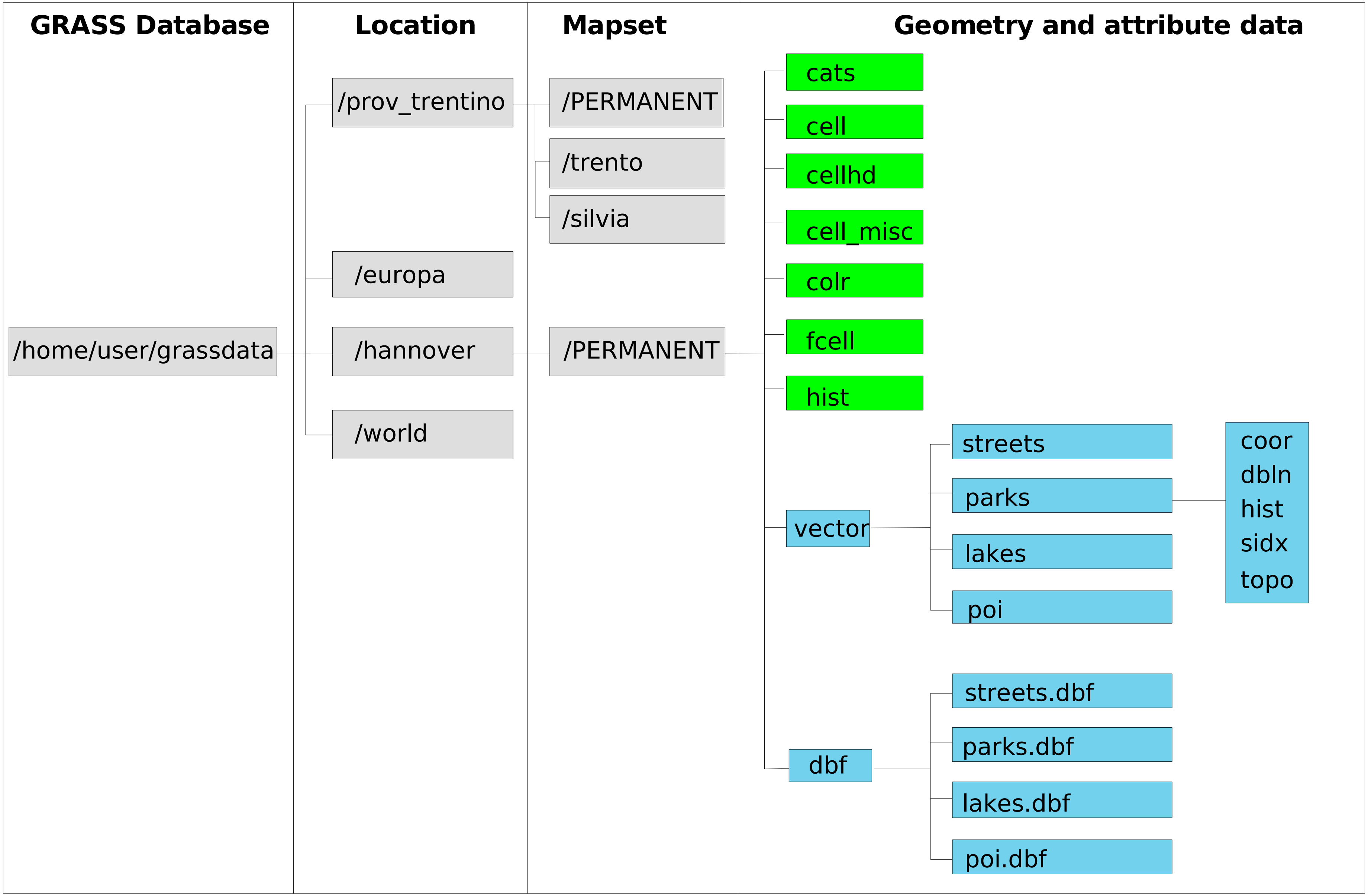
Mapset(s): used to subdivide data organization e.g. by user names, subregions or access rights (workgroups)

PERMANENT: The PERMANENT mapset is a standard mapset which contains the definitions of a location. May also contain general cartography as it is visible to all users

Multi-User support: multiple users can work in a **single location** using different mapsets. Access rights can be managed per user. No user can modify/delete data of other users.

GRASS: Geographic Resources Analysis Support System

Example for Location and Mapsets



Raster map analysis

- DEM analysis
- Raster map algebra
- Geocoding of scanned map
- Volume data processing

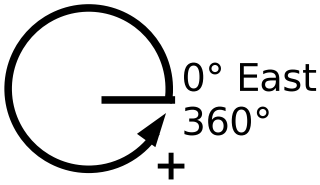
GRASS Command Classes

Prefix	Class	Functionality
d.*	display	graphical output (screen)
r.*	raster	raster data processing
r3.*	raster3D	raster voxel 3D data processing
i.*	imagery	image processing
v.*	vector	vector data processing
g.*	general	general file operations (copy, rename of maps, ...)
m.*	misc	miscellaneous commands
ps.*	postscript	map creation in Postscript format

Raster data analysis: Slope and aspect from DEM

- Calculating slope and aspect from a DEM

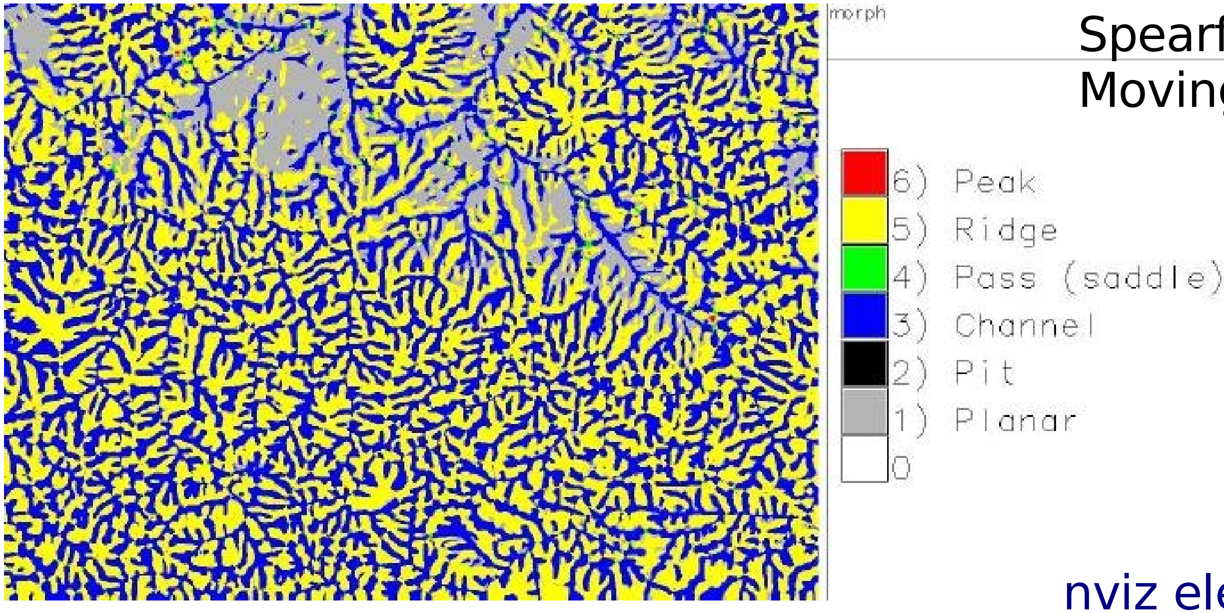
```
# First we reset the current GRASS region settings to the input map:
g.region rast=elevation.10m -p

r.slope.aspect el=elevation.10m as=aspect.10m sl=slope.10m

d.rast aspect.10m
d.rast.leg slope.10m
```



- Note: horizontal angles are counted counterclockwise from the East
- Slopes are calculated by default in degrees
- Also curvatures can be calculated

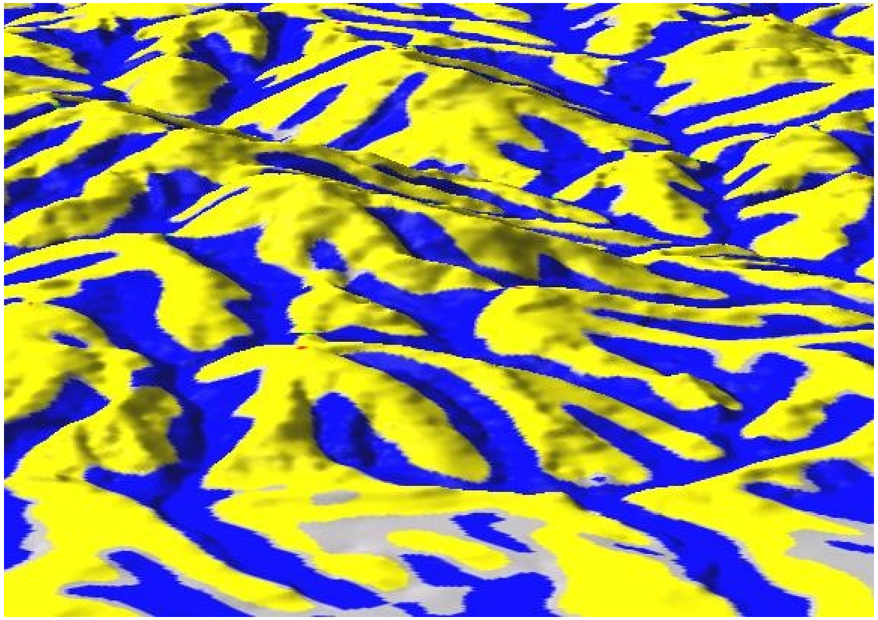
Raster data analysis: Geomorphology

DEM: r.param.scale



Spearfish DEM: 10m
Moving window size: 25x25

nviz elev=elevation.10m col=morph



set region/resolution to the input map:
```
g.region rast=elevation.10m -p
```

generalize with size parameter
```
r.param.scale elevation.10m out=morph \
                     param=feature size=25
```

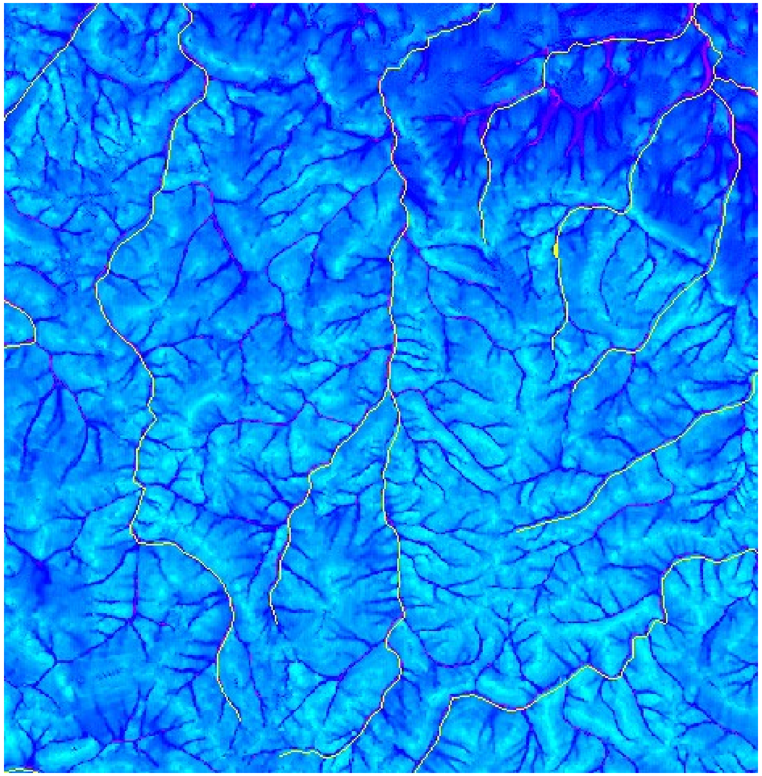
with legend
```
d.rast.leg morph
```

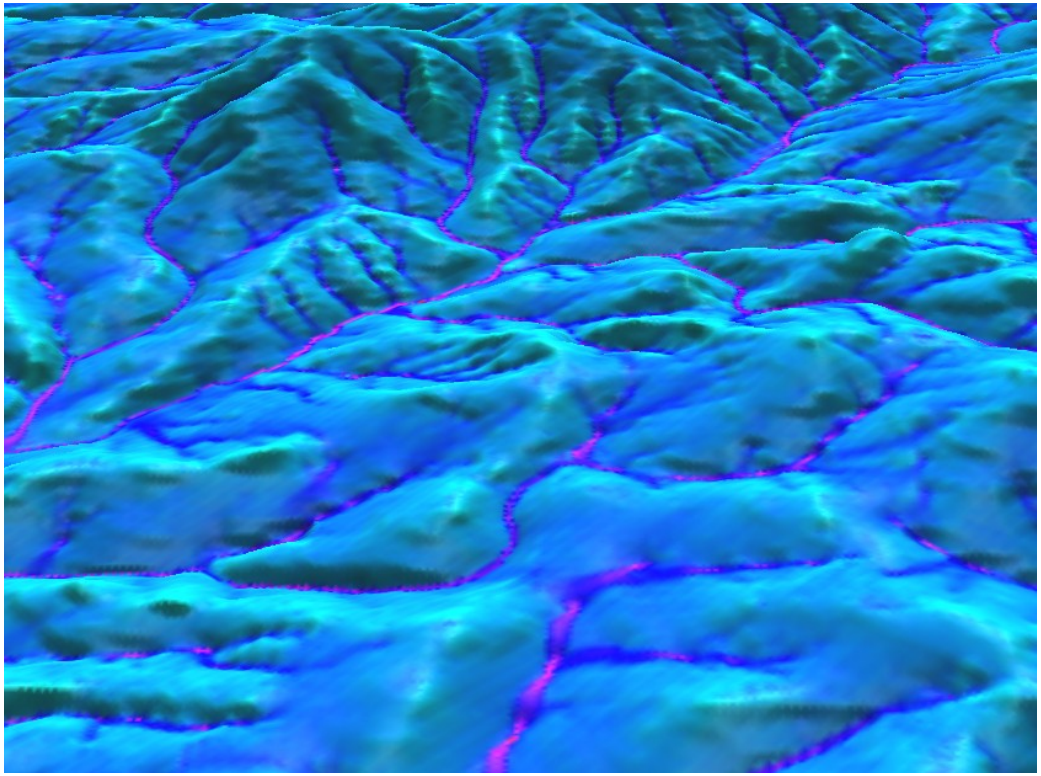
view with aspect/shade map (or QGIS)
```
d.his h=morph i=aspect.10m
```

Raster data analysis: Water flows - Contributing area

Topographic Index: ln(a/tan(beta))

```
g.region rast=elevation.10m -p
r.topidx in=elevation.10m out=ln_a_tanB

d.rast ln_a_tanB
d.vect streams col=yellow
# ... the old vector stream map nicely deviates from the newer USGS DEM

nviz elevation.10m col=ln_a_tanB
```





Raster data analysis: further methods

- Additional DEM analysis modules:
  - depression areas can be filled with **r.fill.dir**
  - flowlines can be calculated with **r.flow**
  - trace a flow through a DEM: **r.drain**
  - watershed analysis can be done with **r.watershed** and **r.terraflow**
  - cost surfaces: **r.cost**

- Energy:
  - cast shadows, astronomical calculations of sun position: **r.sunmask**
  - energy budget: **r.sun**

- Line of sight:
  - viewsheds can be generated with: **r.los**

- Interpolation methods
  - 2D inverse distance weighted: **v.surf.idw**
  - 2D from contour lines: **r.surf.contour**
  - 2D bilinear: **r.bilinear**
  - 2D regularized splines with tension (with cross validation): **v.surf.rst**
  - 3D regularized splines with tension (with cross validation): **v.vol.rst**
  - 2D/3D kernel densities: **v.kernel**

  - via R-stats: kriging, predictive models etc

Raster map algebra

- A powerful raster map algebra calculator is **r.mapcalc**
  See for functionality:

```
g.manual r.mapcalc &
```

- With a simple formula we filter all pixels with elevation higher than 1500m from the Spearfish DEM:

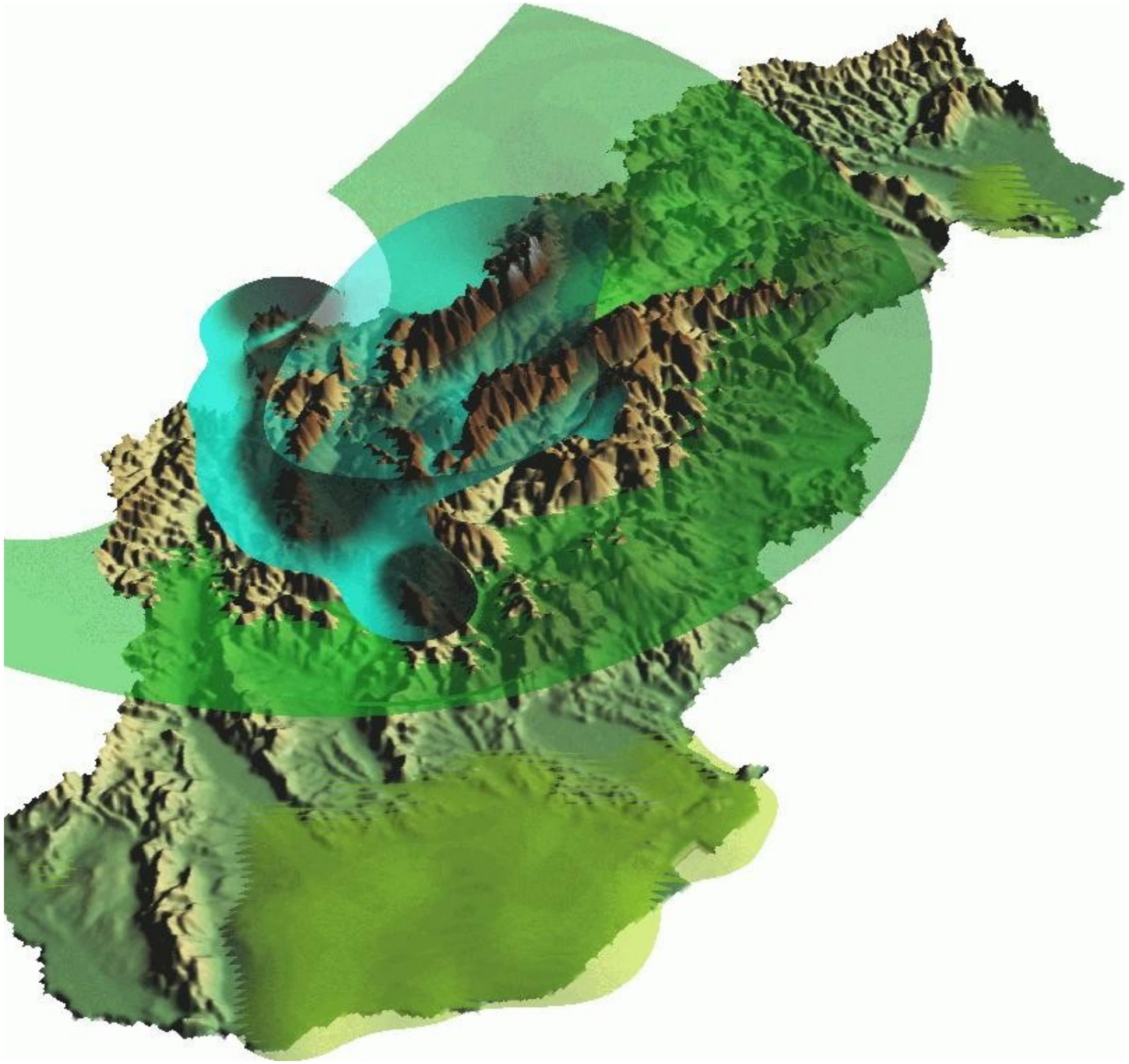
```
r.mapcalc "elev_1500 = if(elevation.dem > 1500.0, elevation.dem, null())"
d.rast elev_1500

d.rast aspect
d.rast -o elev_1500
```

Volume map processing: Demo

GRASS was enhanced to process and visualize Volumes (consisting of 3D voxels)

Functionality:

- 3D import/export
- 3D Regularized Splines with Tension interpolation
- 3D map algebra
- NVIZ volume visualization: Isosurfaces and Profiles



Working with vector data

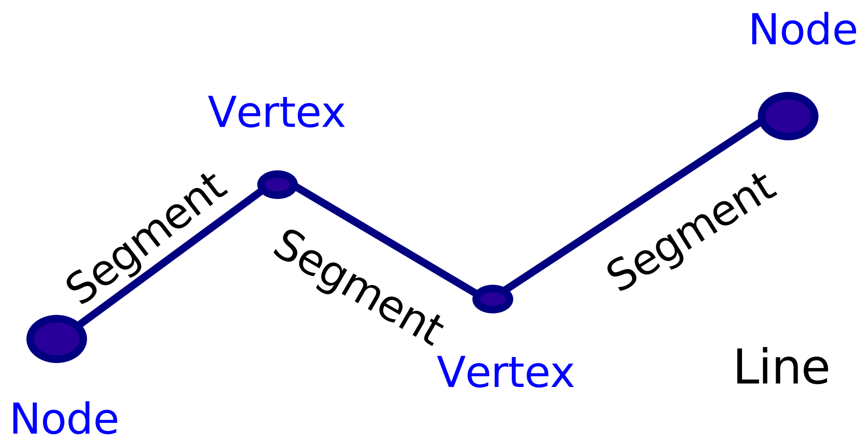
- Vector map import
- Attribute management
- Buffering
- Extractions, selections, clipping, unions, intersections
- Conversion raster-vector and vice verse
- Digitizing in GRASS and QGIS
- Working with vector geometry

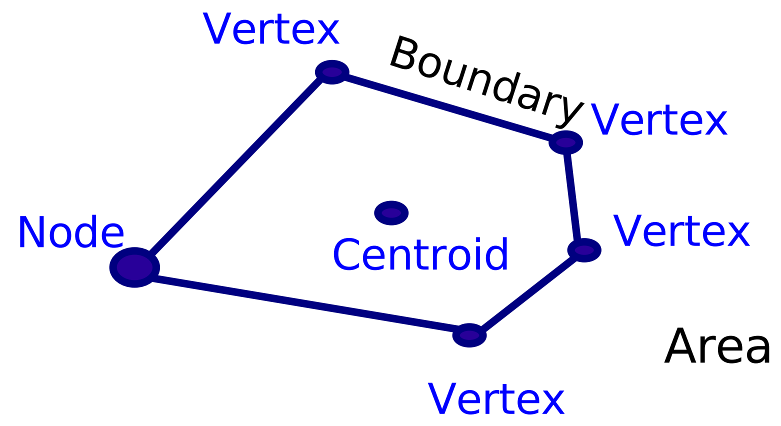
GRASS 6 Vector data

Vector geometry types

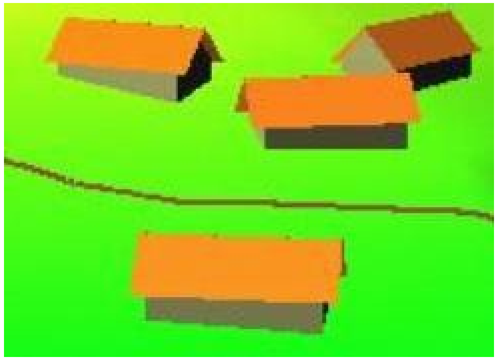
- Point
- Centroid
- Line
- Boundary
- Area (boundary + centroid)
- face (3D area)
- [kernel (3D centroid)]
- [volumes (faces + kernel)]

Geometry is **true** 3D: x, y, z





Faces



Raster-Vector conversion – extraction 1/2

Extraction of residential areas from raster landuse map

set current region to map; look at the landuse/landcover map with legend:

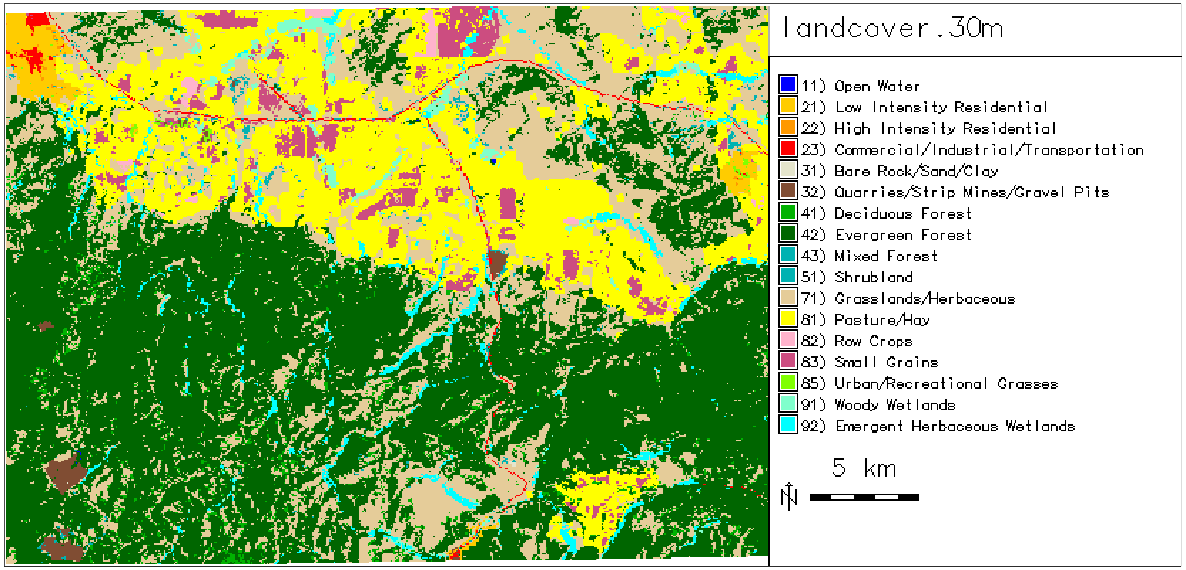
```
g.region rast=landcover.30m -p
d.erase
d.rast.leg -n landcover.30m
```

Automated vectorization of the landuse/landcover map:

```
r.to.vect -s landcover.30m out=landcover30m feature=area
```

see attribute table ('-p' prints the current connection between vector
geometry and attribute table – note that GRASS can link to various DBMS):

```
v.db.connect -p landcover30m       # ... will tell you that it is a DBF table
v.db.select landcover30m
```



Raster-Vector conversion – extraction 2/2

Extraction of residential areas from raster landuse map

generate list of unique landuse/landcover types from text legend output:

```
v.db.select landcover30m | sort -t '|' -k2 -n -u
```

This pipe '|' character is a nice way of combining Unix commands. The output of the first command is sent into the second and so forth...

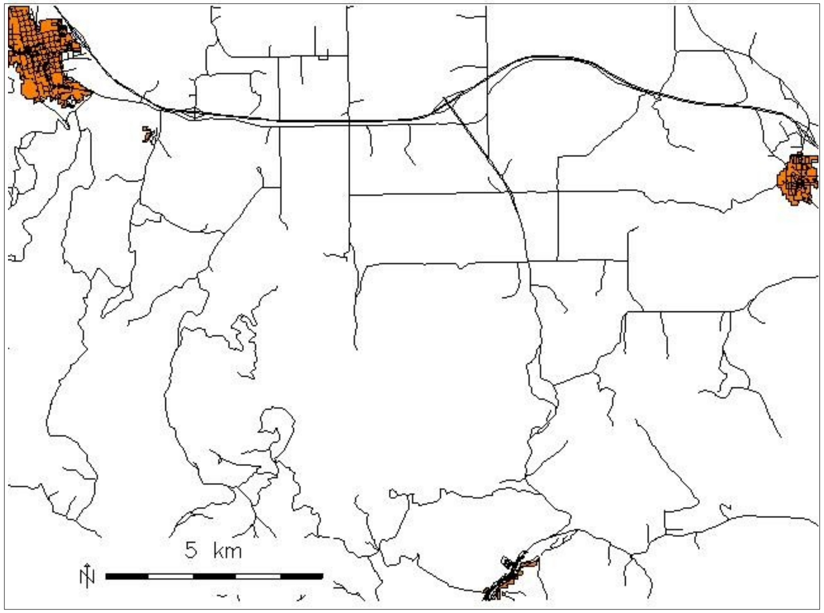
sort is here sorting by second column on numbers (-n) and extracts unique (-u) rows only

#display selected categories:

```
d.erase
d.vect landcover30m \
     where="value=21 or value=22" \
     fcol=orange
```

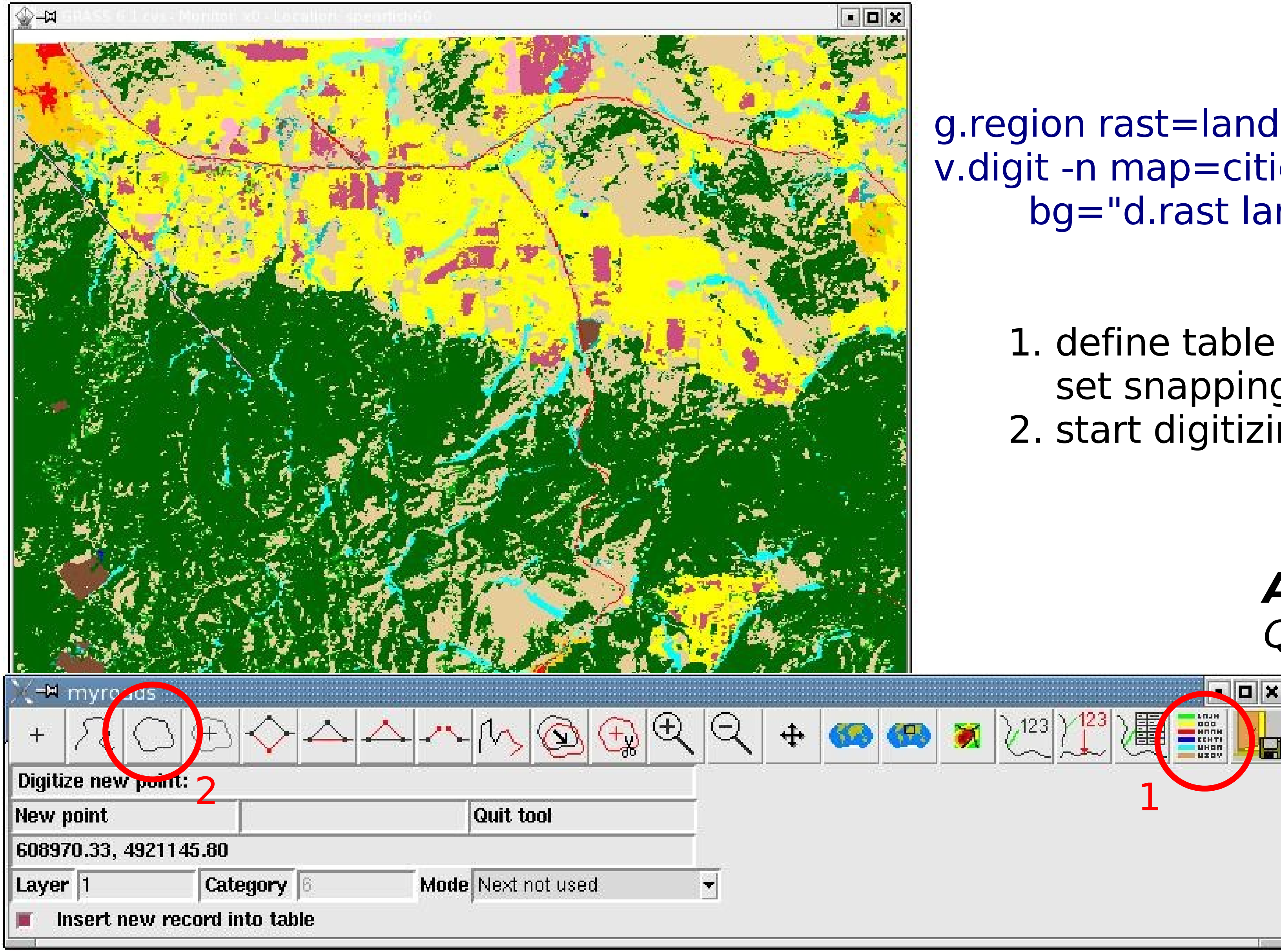
Extract residential area into a new vector map:

```
v.extract landcover30m out=residential where="value=21 or value=22"

d.frame -e
d.vect residential fcol=orange \
          type=area
d.vect roads
d.barscale -mt
```



Creating/modifying vector maps

Digitizing in GRASS




```
g.region rast=landcover.30m -p
v.digit -n map=cities \
      bg="d.rast landcover.30m"
```

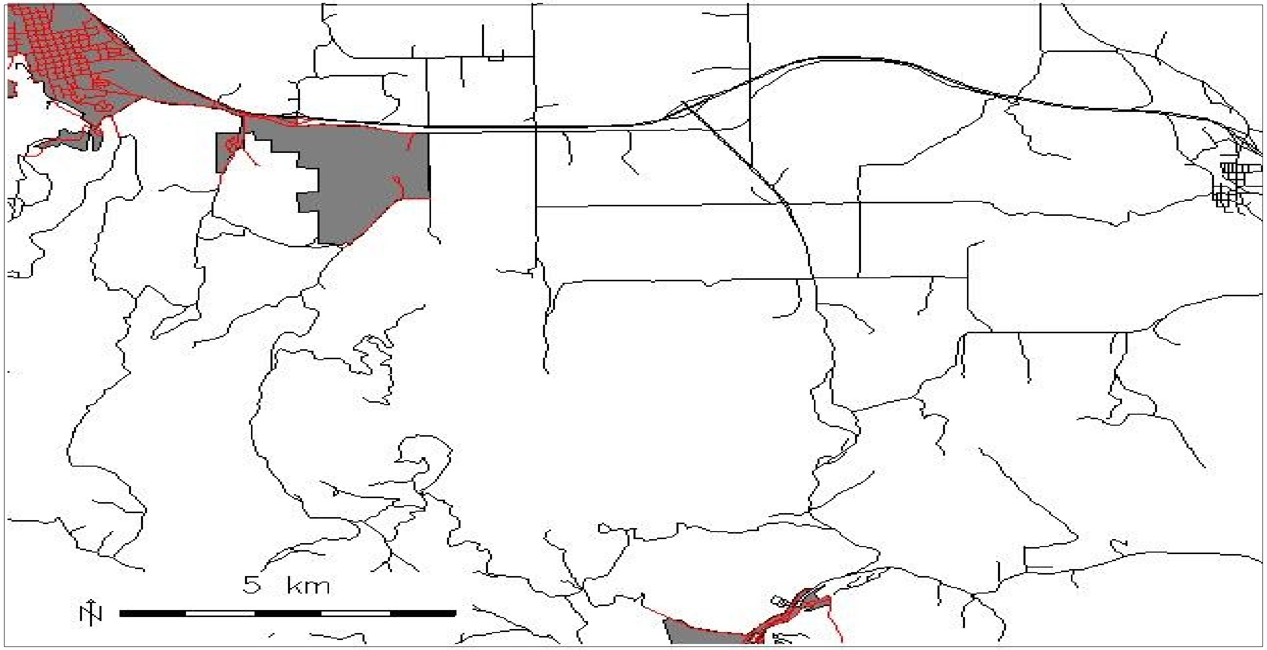
1. define table
   set snapping threshold
2. start digitizing

Alternative:
QGIS digitizer!

Vector map clipping

Selection example: Roads in urban areas

```
# display roads and residential areas:
 d.erase
 d.vect roads
 d.vect residential

# extract all roads within the urban areas:
 v.select ain=roads bin=residential out=urban_roads
 d.vect urban_roads col=red
```

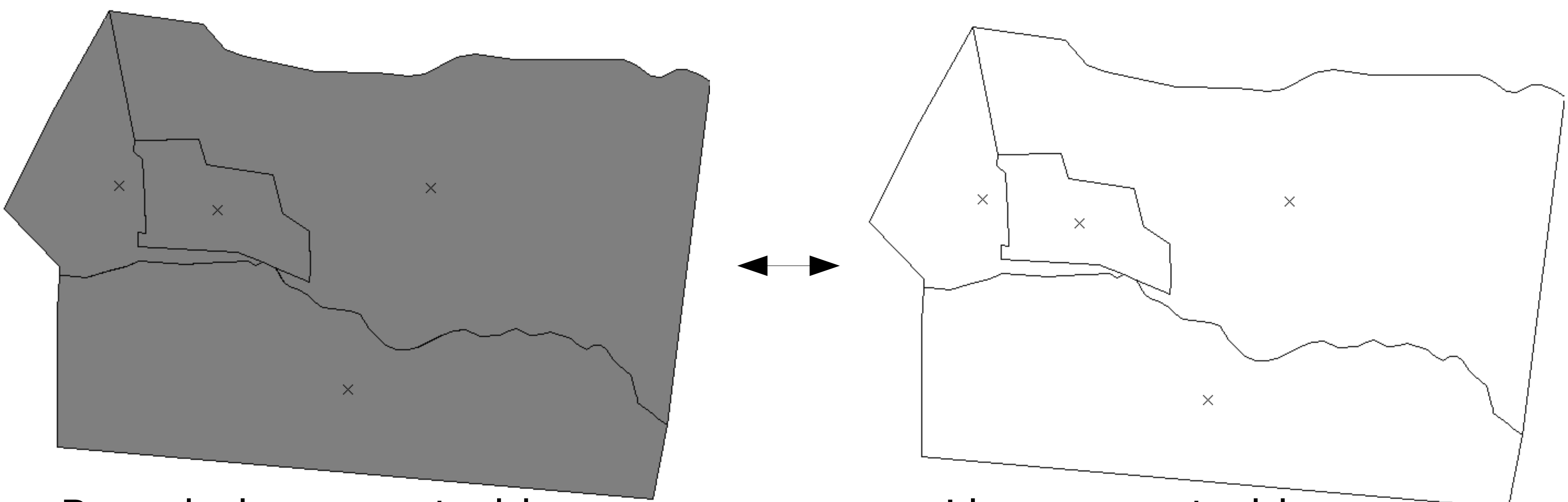


Changing vector types

In GRASS an **area** polygon is defined by a boundary + a centroid.
Lines can be a (poly)line or a boundary.

Vector types can be changed by **v.type/v.build.polyline** such as

point	↔	centroid
3D point	↔	kernel (3D centroid)
line	↔	polyline
line	↔	boundary
3D area	↔	face



Boundaries + centroids   Lines + centroids

Vector networking

- Overview
- Shortest path analysis

Vector network analysis methods

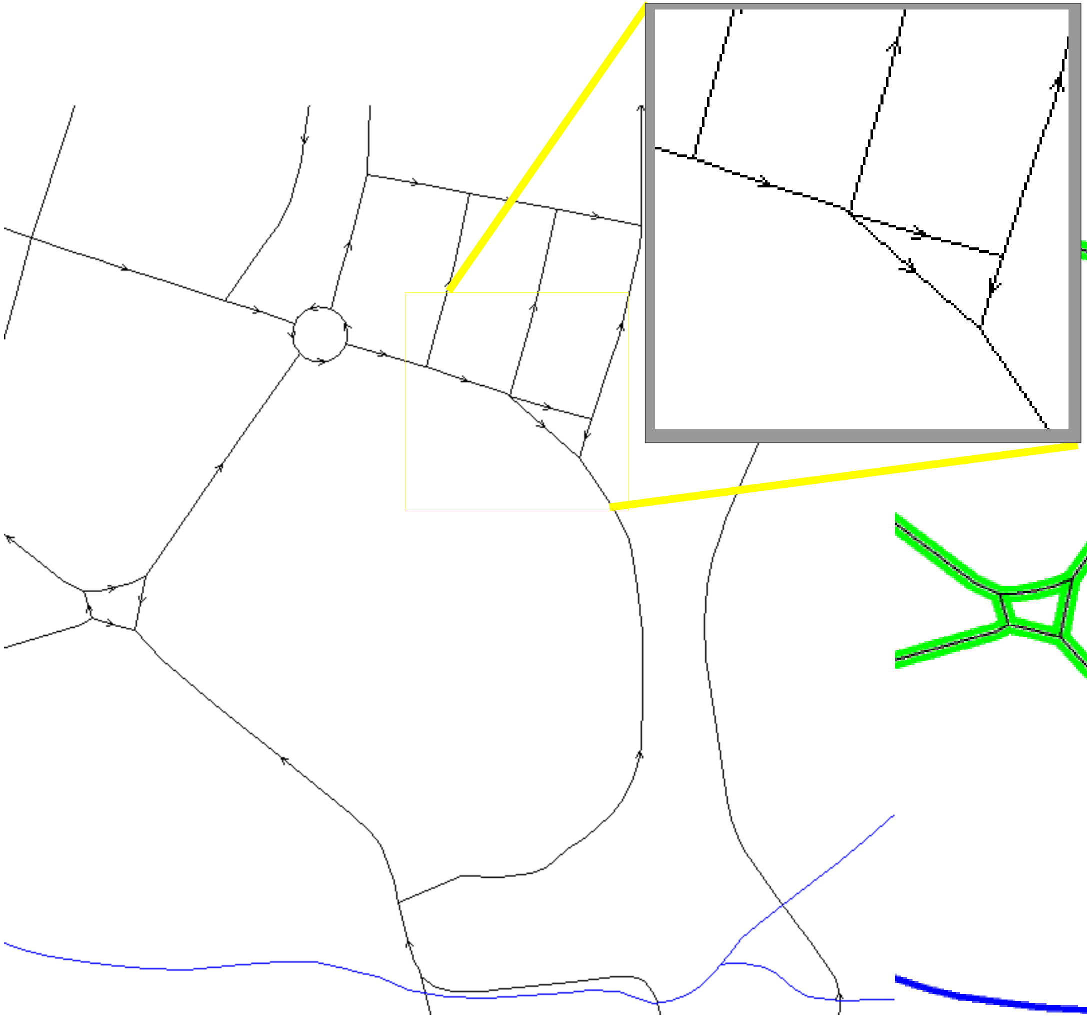
Available methods:

- find shortest path along vector network - *road navigation*
- find optimal round trip visiting selected nodes (Traveling salesman problem) - *delivering of goods*
- find optimal connection between nodes (Minimum Steiner tree) - *ADSL network*
- subdivide a network in subnetworks (iso distances) - *how far can I go from a node in all directions*
- find subnetworks for set of nodes (subnet allocation) - “catchment area” for fire brigade etc

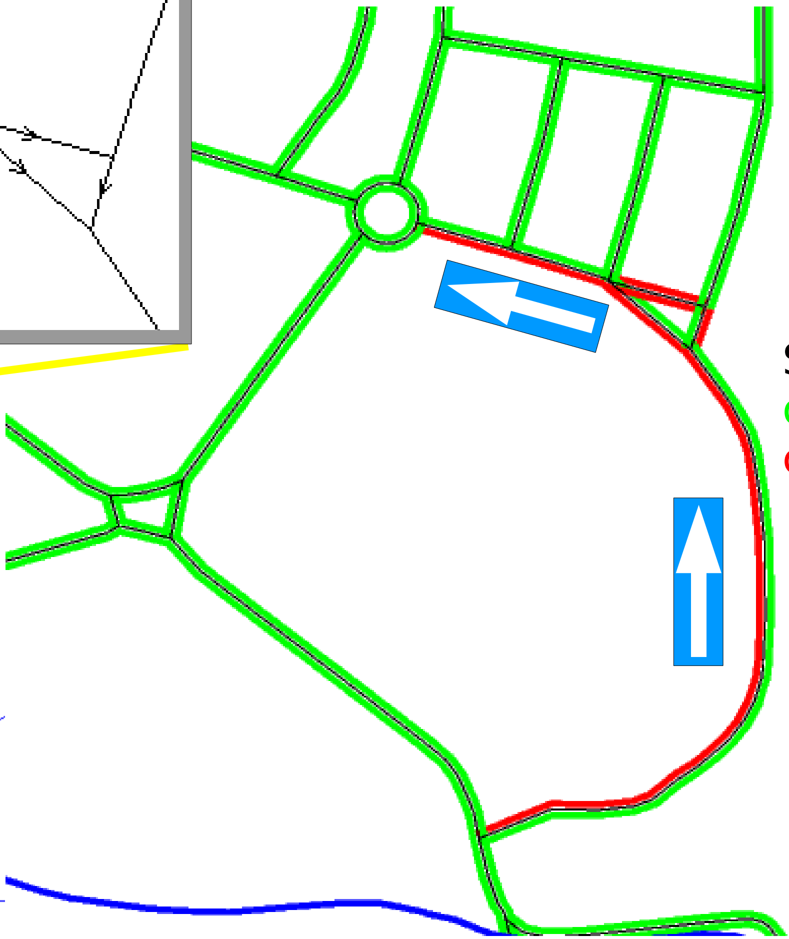
Vector network analysis methods

Vector network with one way roads

drawn in ps.map



Generic vector directions



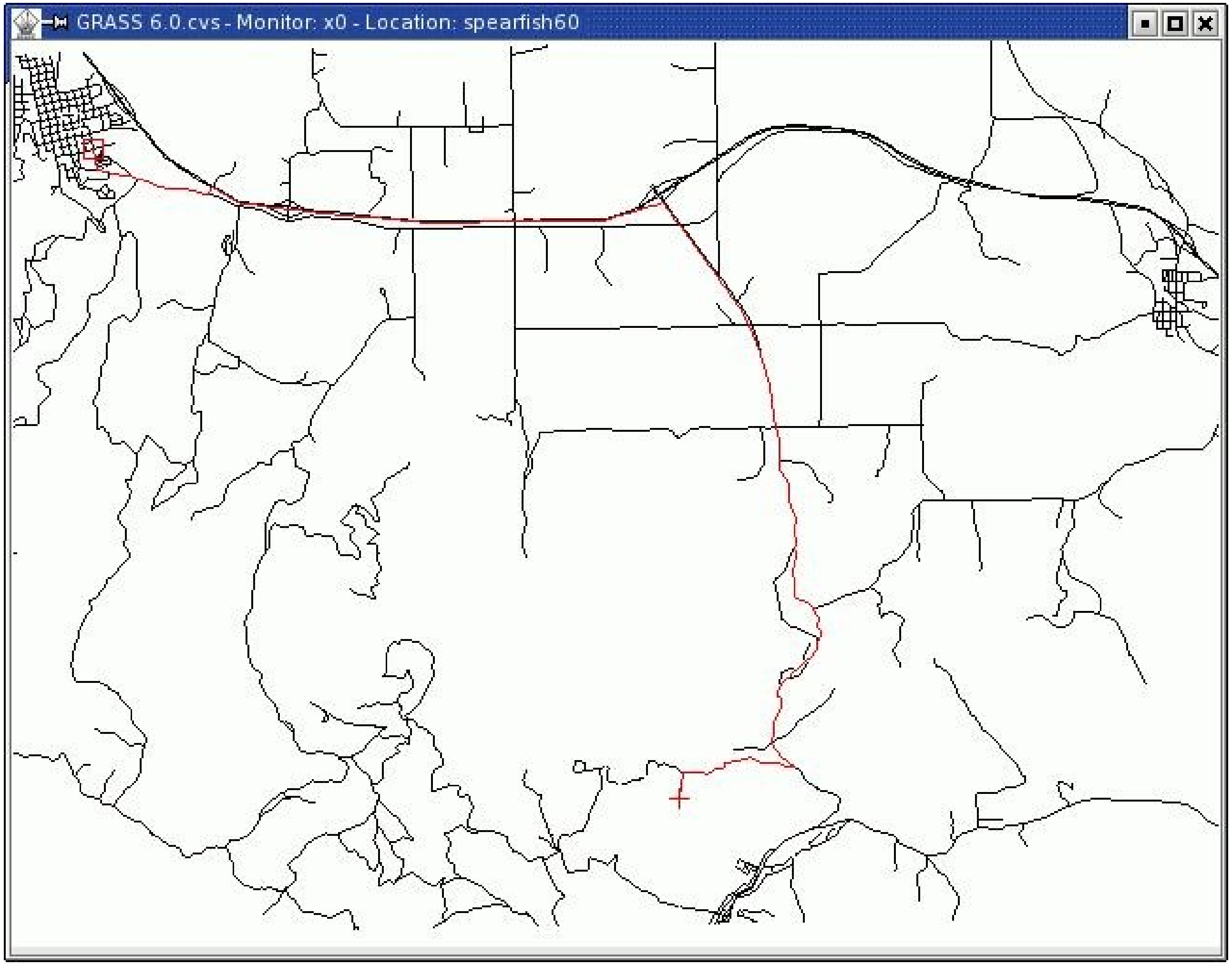
One attribute column for each direction
Value -1 closes direction (for one way streets)

Vector networking

Shortest path with **d.path**

d.vect roads
d.path roads

or:
v.net.path

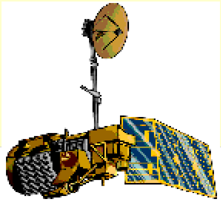


Further vector network exercises:
http://mpa.itc.it/corso_dit2004/grass04_4_vector_network_neteler.pdf

Working with own data - Import/Export/Creating Locations

- Import of LANDSAT-7 data
- Creating a new location external data files
- Creating from EPSG code/interactively a new location

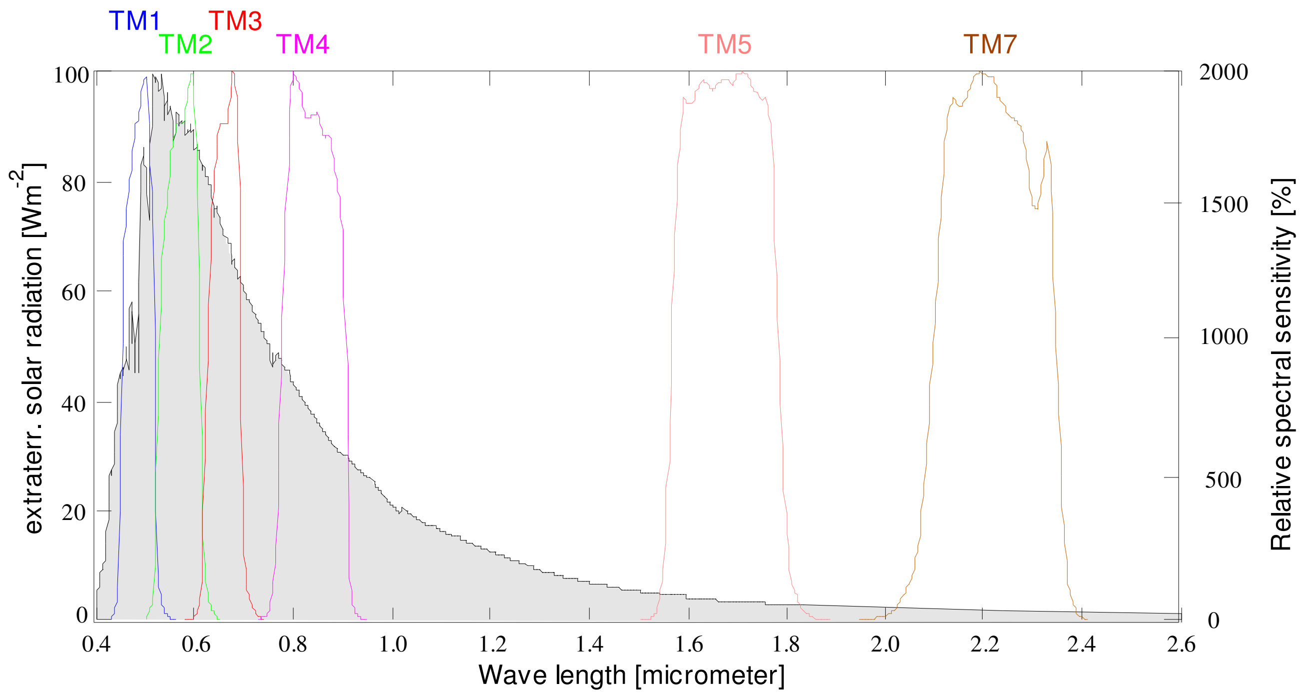
Import of LANDSAT-7 Erdas/Img raster maps 1/2



- A LANDSAT-7 scene has been prepared (reprojected, spatially subset):

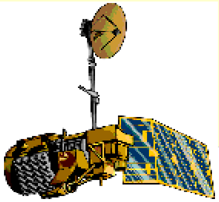
  - spearfish_landsat7_NAD27_vis_ir.img:
    TM10,TM20,TM30 (blue, green, red), TM40 (NIR), TM50, TM70 (MIR)
  - spearfish_landsat7_NAD27_tir.img:
    TM62 (TIR low gain), TM62 (TIR high gain)
  - spearfish_landsat7_NAD27_pan.img:
    TM80 (panchromatic)

Neteler/Mitasova 2002



Solar spectrum and LANDSAT channels (thermal channel 6 not shown)

Import of LANDSAT-7 Erdas/Img raster maps 2/2



- The import is done with **r.in.gdal**:

```
r.in.gdal -e in=spearfish_landsat7_NAD27_vis_ir.img out=tm
# To keep the numbering right, we rename tm.6 to the
# correct number tm.7:
g.rename rast=tm.6,tm.7

r.in.gdal -e in=spearfish_landsat7_NAD27_tir.img out=tm6

r.in.gdal -e in=spearfish_landsat7_NAD27_pan.img out=pan
```

- Generate a RGB composite on the fly (zoom to map first):

```
g.region rast=tm.1 -p
d.rgb b=tm.1 g=tm.2 r=tm.3
```

You should see the Spearfish area in near-natural colors.

Creating new GRASS locations

- Both **r.in.gdal** and **v.in.ogr** offer a **location=** parameter to create a new location from the import dataset's metadata

  Example:
  r.in.gdal -e in=spearfish_landsat7_NAD27_tir.img out=tm6 location=utm13

- Launching GRASS (again) permits to
  - create a new location from EPSG code
  - create a new location interactively
- See the workshop handout for details

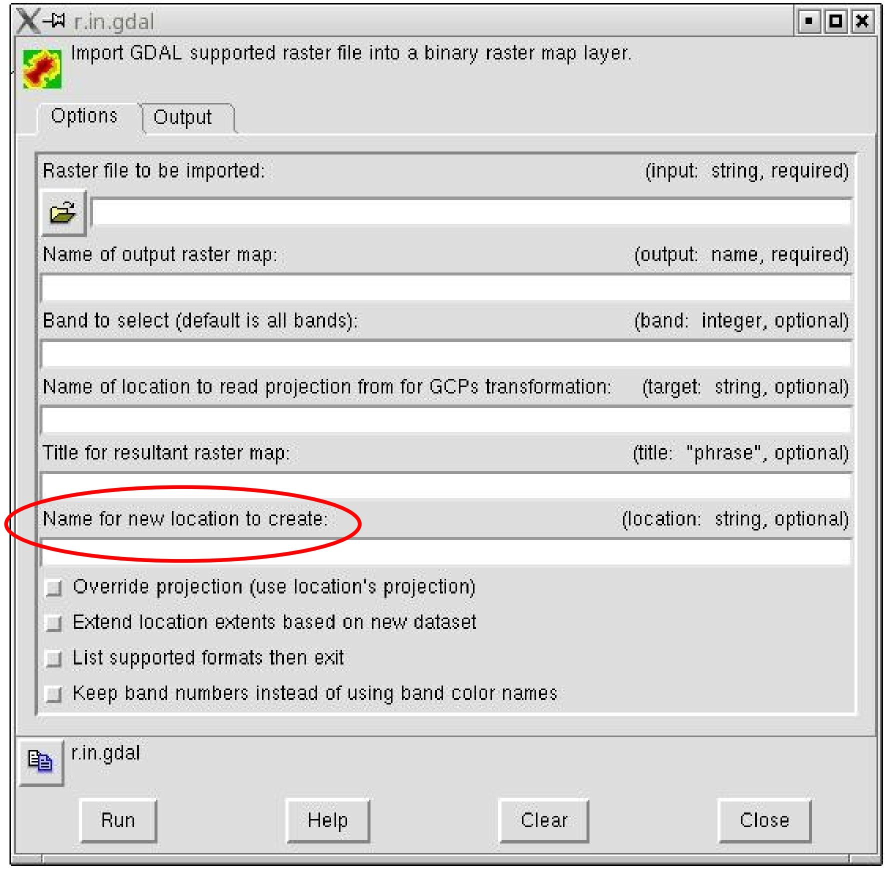


Image processing

- Image classification
- Image fusion with Brovey transform
- Natural color composites
- Calculating a degree Celsius map from the LANDSAT thermal channel

Import of LANDSAT-7 Erdas/Img

Image Classification

Unsupervised & Supervised Image Classification

- classification methods in GRASS:

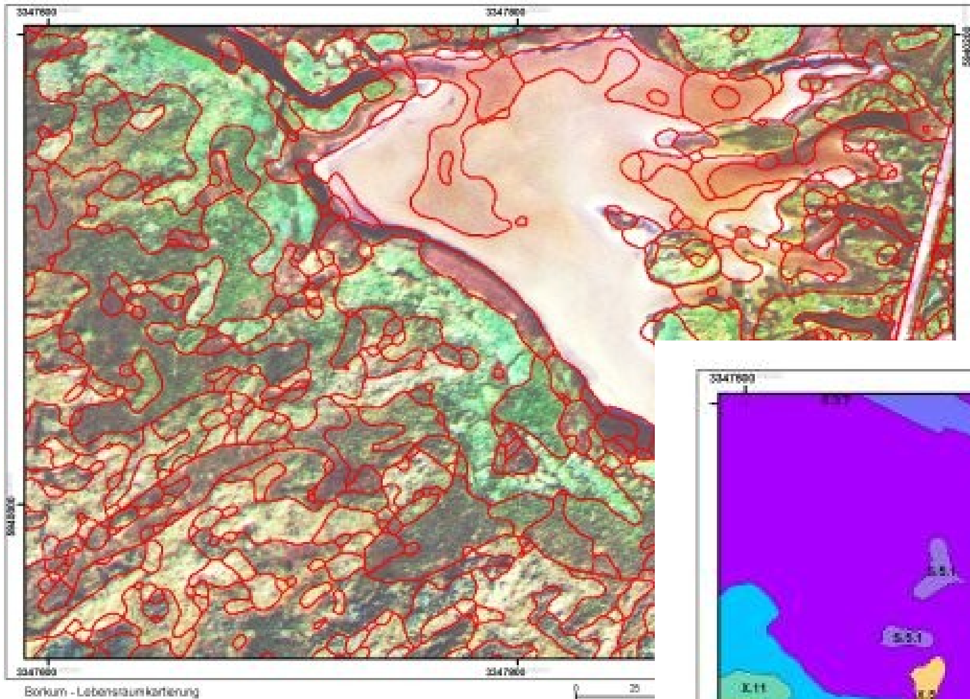
	radiometric, unsupervised	radiometric, supervised		radio- and geometric supervised
Preprocessing	i.cluster	i.class (monitor)	i.gensig (maps)	i.gensigset (maps)
Computation	i.maxlik	i.maxlik	i.maxlik	i.smap

- all image data must be first listed in a group (i.group)

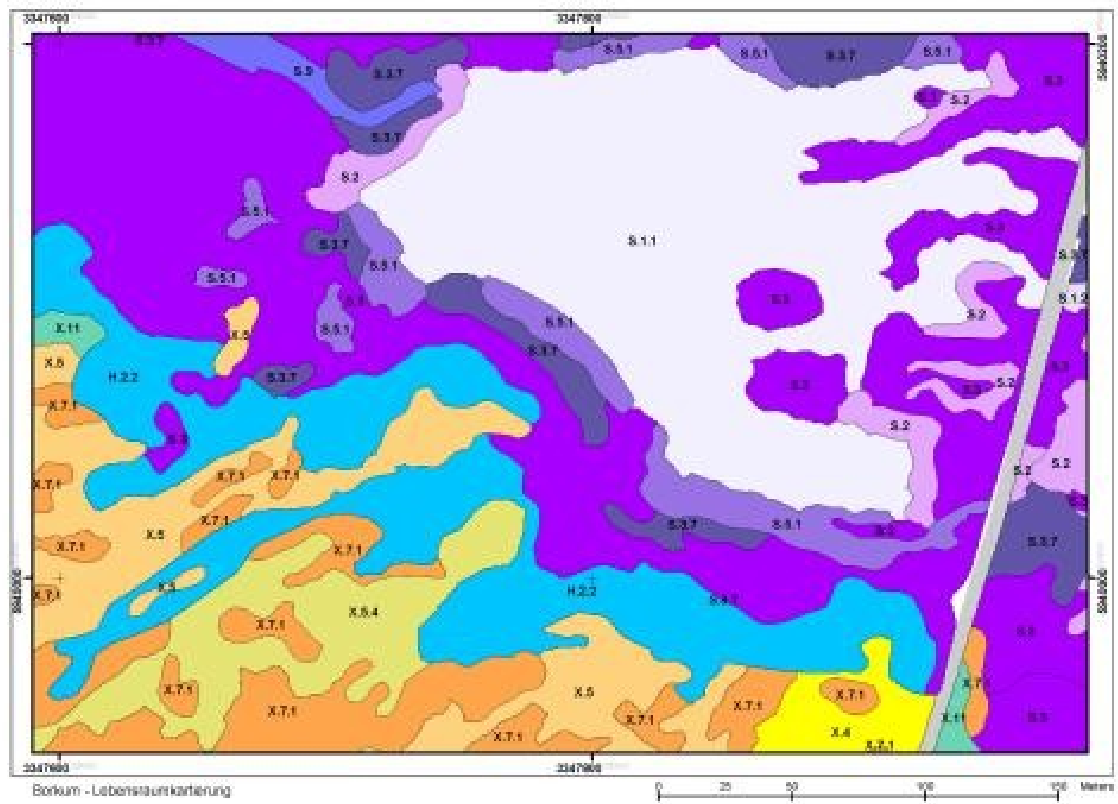
- See handout for unsupervised classification example

GRASS: Geographic Resources Analysis Support System

Image classification



GRASS supports
- Image geocoding and ortho-rectification
- Analysis of aerial and satellite data
- Time series analysis

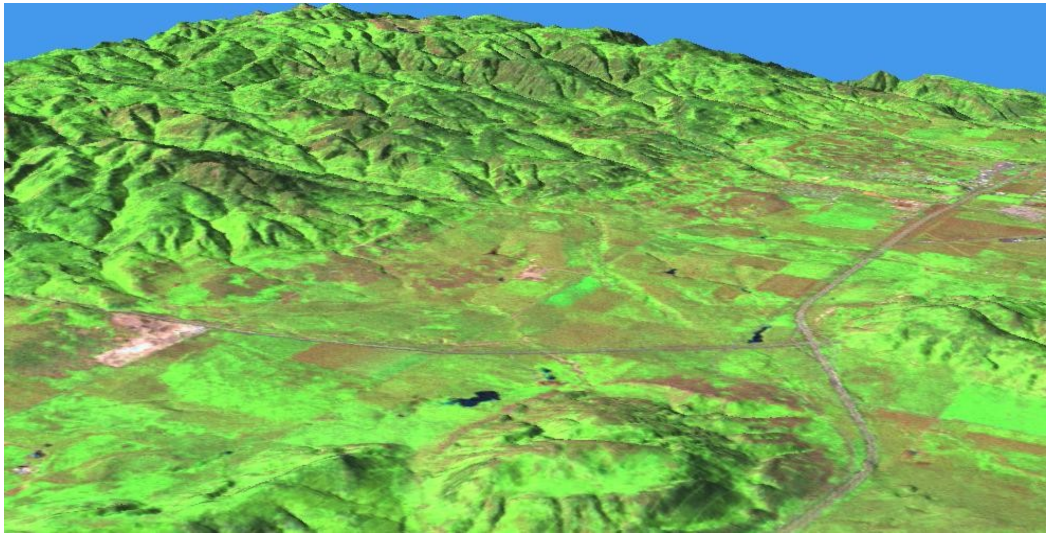


- Biotope monitoring from digital aerial cameras (HRSC-X and DMC)
- SMAP Classifier of GRASS

Image fusion: Brovey transform

We use the earlier imported LANDSAT-7 scene to perform image fusion of the channels 2 (red), 4 (NIR), and 5 (MIR):

```
g.region -dp
i.fusion.brovey -l ms1=tm.2 ms2=tm.4 ms3=tm.5 pan=pan out=brovey

# zoom to fused channel
g.region -p rast=brovey.red

# color composite:
r.composite r=brovey.red g=brovey.green b=brovey.blue n out=tm.brovey

d.rast tm.brovey

nviz elevation.10m col=tm.brovey

# Increase visual resolution in NVIZ
# with Panel -> Surface
#                  -> Polygon resolution
#                       (lower! the value)
```



Natural color composites: LANDSAT-7 RGB

The i.landsat.rgb script performs a histogram-area based color optimization:

Standard RGB

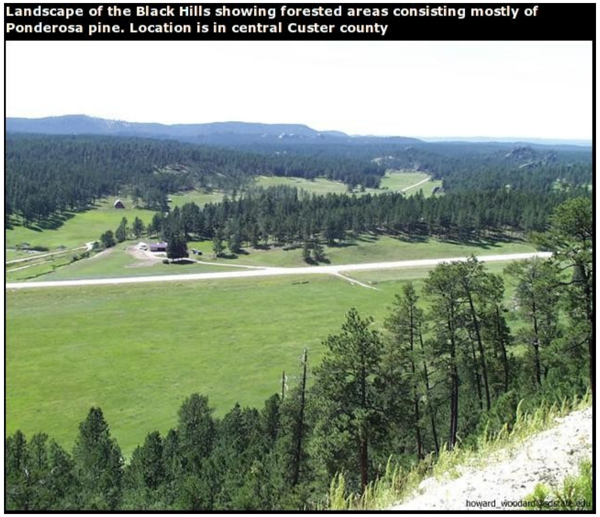


Photo: H.J. Woodard, SD Stae Univ.

http://plantsci.sdstate.edu/woodardh/Soils_and_Ag/Black_Hills/Soil_Characteristics_Profiles/landscape_pine.htm

Enhanced RGB

Recalibrating the LANDSAT-7 thermal channel 1/2

TM61: Conversion of temperature first to Kelvin, then to degree Celsius

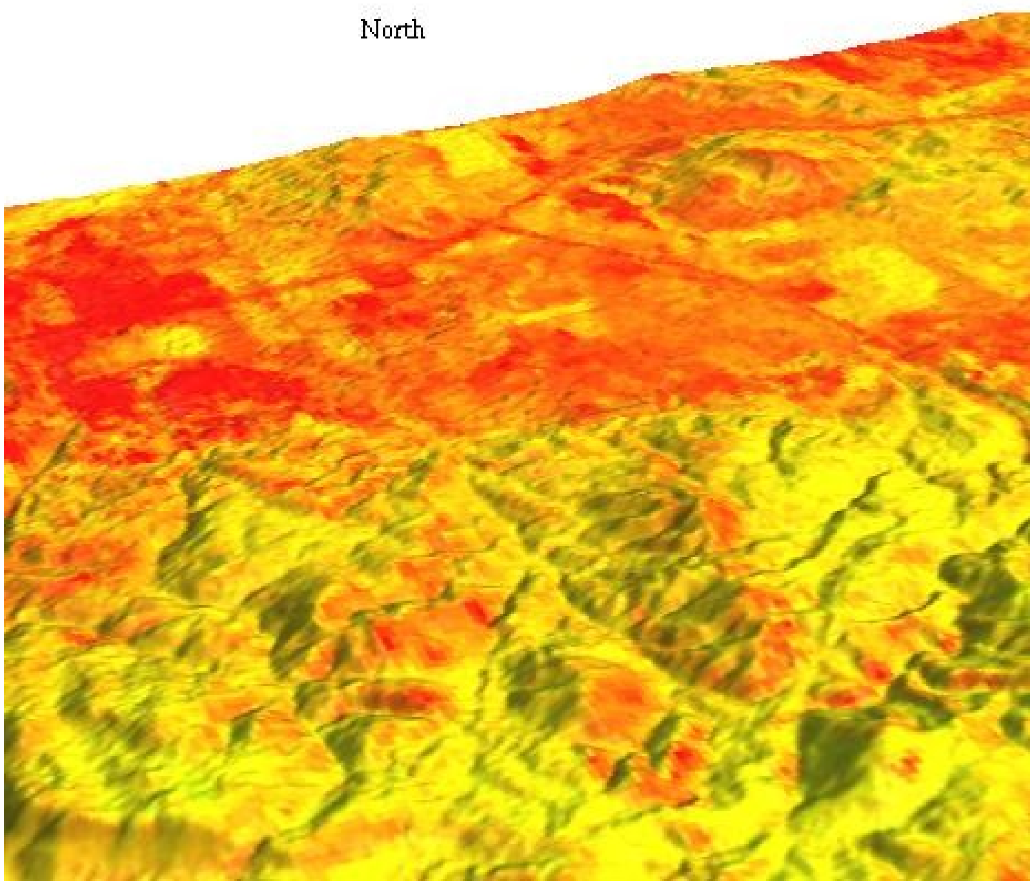
```
g.region rast=tm6.1 -p

#DN: digital numbers (coded temperatures)
 r.info -r tm6.1
   min=131
   max=175

# Conversion of DN to spectral radiances:
 r.mapcalc "tm61rad=((17.04 - 0.)/(255. - 1.))*(tm6.1 - 1.) + 0."
 r.info -r tm61rad
   min=8.721260
   max=11.673071

# Conversion of spectral radiances to absolute temperatures (Kelvin):
# T = K2/ln(K1/L_l + 1))
 r.mapcalc "temp_kelvin=1260.56/(log (607.76/tm61rad + 1.0))"
 r.info -r temp_kelvin
   min=296.026722
   max=317.399879
```

Recalibrating the LANDSAT-7 thermal channel 2/2

Note: Land surface temperatures are not air temperatures!
LANDSAT passes at around 9:30 local time

TM61: ... conversion to degree Celsius

```
# We currently have the land surface temperature map in Kelvin.
# Conversion to degree Celsius:
 r.mapcalc "temp_celsius=temp_kelvin - 273.15"
 r.info -r temp_celsius
   min=22.876722
   max=44.249879

# New color table:
r.colors temp_celsius col=rules << EOF
-10 blue
15 green
25 yellow
35 red
50 brown
EOF

d.rast.leg temp_celsius

g.region rast=elevation.dem -p
nviz elevation.dem col=temp_celsius
```



GRASS and geostatistics

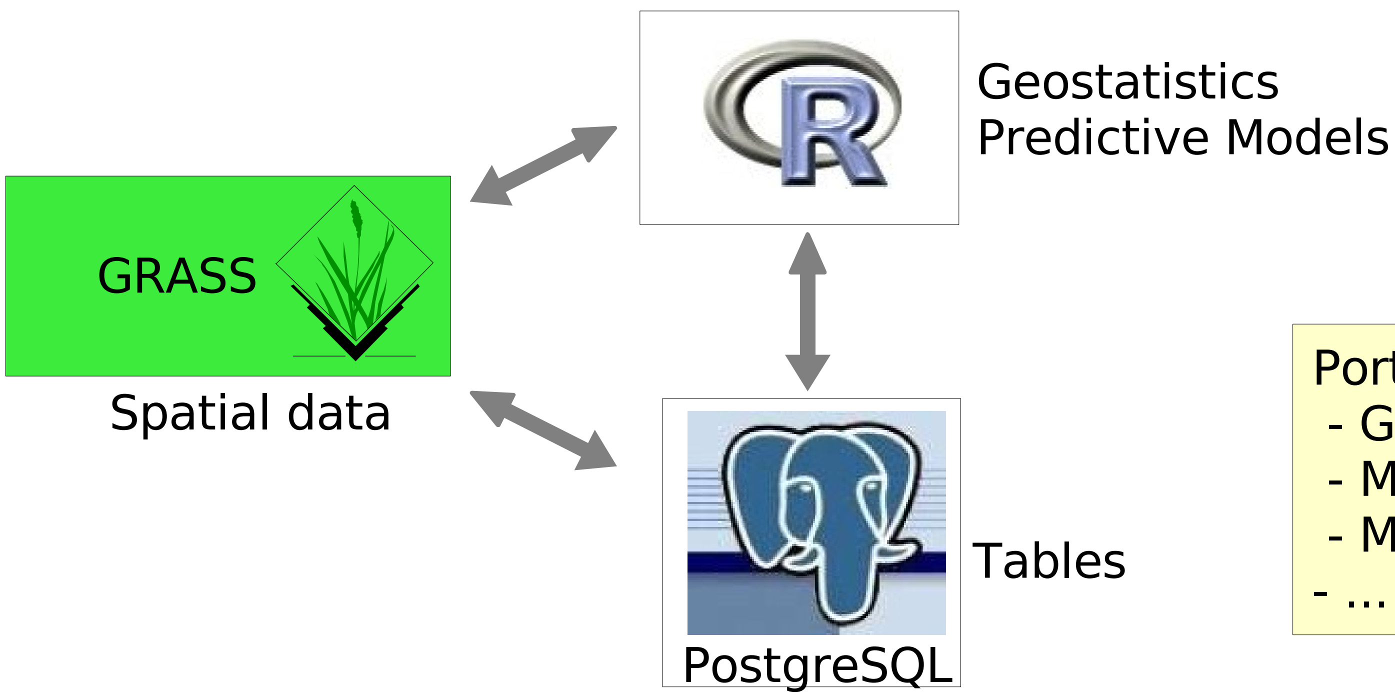
- R-stats/GRASS interface

GRASS/R-stats interface - R-stats/PostgreSQL interface

http://www.r-project.org
http://grass.itc.it/statsgrass/

- R-stats is a powerful statistical language
- Spatial extentions available for all kinds of **geostatistics, spatial pattern analysis, time series** etc
- Interface to exchange raster and point data between GRASS and R-stats

- Rdbi: connects R-stats to PostgreSQL



Portable Applications:
- GNU/Linux
- MS-Windows
- MacOSX
- ...

GRASS/R-stats interface

R statistical language

Web site and CRAN:
http://www.r-project.org
http://cran.r-project.org

Object oriented statistical language, a “S” dialect. Examples:

```
 R
> 1
[1]

> 1+1
[1] 2

# assignment into object:
> x <- 1+1
> x
[1] 2

> q()
Save workspace image [y/n/c] n
```

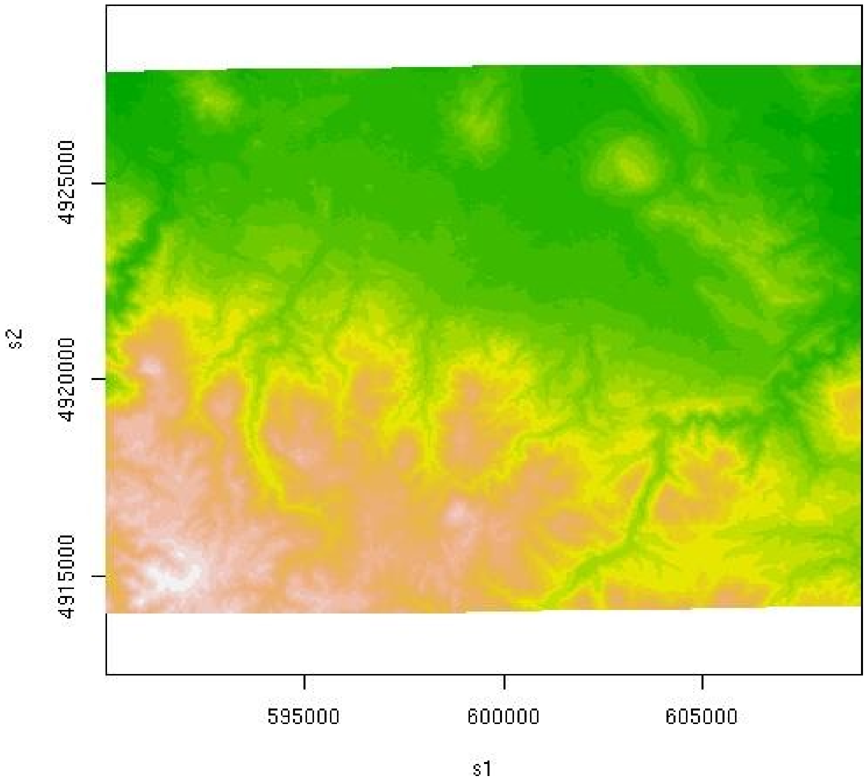
GRASS/R-stats interface

GRASS 6 and R statistical language

```
grass61

#reset region:
g.region rast=elevation.dem -p

#in GRASS start R:
R
library(spgrass6)

#load GRASS environment into R:
G <- gmeta6()

#see environment metadata:
str(G)


# Now R is ready for GRASS data analysis.
```

GRASS/R-stats interface

GRASS 6 and R statistical language (cont'd)



get online help:
?readCELL6sp

# get full help:
help.start()

# load GRASS DEM into R:
elev <- readCELL6sp("elevation.dem")

# show map metadata:
summary(elev)

# show map:
image(elev, col=terrain.colors(20))

# leave R:
q()

you have the option to save the current R state to local
file when leaving the program.

GRASS: User map

Who is using GRASS?

AMTI/NASA Ames Research Center USA
Austrian Institute for Avalanche and Torrent Research
Bank of America
Bombardier Aerospace Canada
Brenner Railway Austria
BR-NetProduction (Bavarian Television) Germany
Canadian Forest Service
CEA Monte Bondone
Census USA
CERN Switzerland
CICESE Mexico
CNR Italia
Colorado State University
Comune di Prato, Italy
Comune Milano, Italy
Comune Modena, Italy
Comune di Torino, Italy
Cornell University USA
CSIRO Australia
Deutsche Bank Germany
DLR Germany
Dubai Municipality
DuPont Spain
EDF France
Ericsson Sweden
ETH Zuerich Switzerland
FED USA
Finnish Meteorological Institute
Forschungszentrum Juelich Germany
Forschungszentrum Karlsruhe Germany
GFZ Potsdam Germany
Global Environmental Technology Nigeria Limited
Graz Technical University Austria
Harvard University
Hokkaido University
HPCC NECTEC Bangkok Thailand
Iceland Forest Service Iceland
Inst.of Earthquake Engineering & Seismology (ITSAK) Greece
ISMAA - Centro Agrometeorologico, Istituto Agrario San Michele
JPL NASA
JSC NASA

Landesmuseum Linz Austria
La Poste France
Lawrence Laboratories USA
Lockheed Martin Space USA
Los Alamos National Laboratory
Meteo Poland
MIT Lincoln Laboratory
Nanjing University
National Botanic Garden of Belgium
National Museum Japan
National Radio Astronomy Observatory USA
National Research Center of Soils USA
NCSA Illinois USA
NCSU USA
NIMA USA
NOAA USA (GLOBE DEM generated with GRASS)
NRSA USA
Onera France (running SPOT etc.)
Politecnico di Milano
Politecnico di Torino
Princeton University
Procergs Brasilia

Purdue University
Qualcomm USA
Regione Toscana
Rutgers University
Sevilla University Spain
South African Weather Bureau (METSYS)
Stockholm Environment Institute-Boston
Teledetection France
Telefónica Spain
TU Berlin
TU Muenchen
UC Davis
UFRGS Brasilia
University of Costa Rica
University of Sydney
University of Toronto Canada
University of Trento, Italy
US Army
US Bureau of Reclamation
US Dep. of Agriculture
VA Linux Systems USA

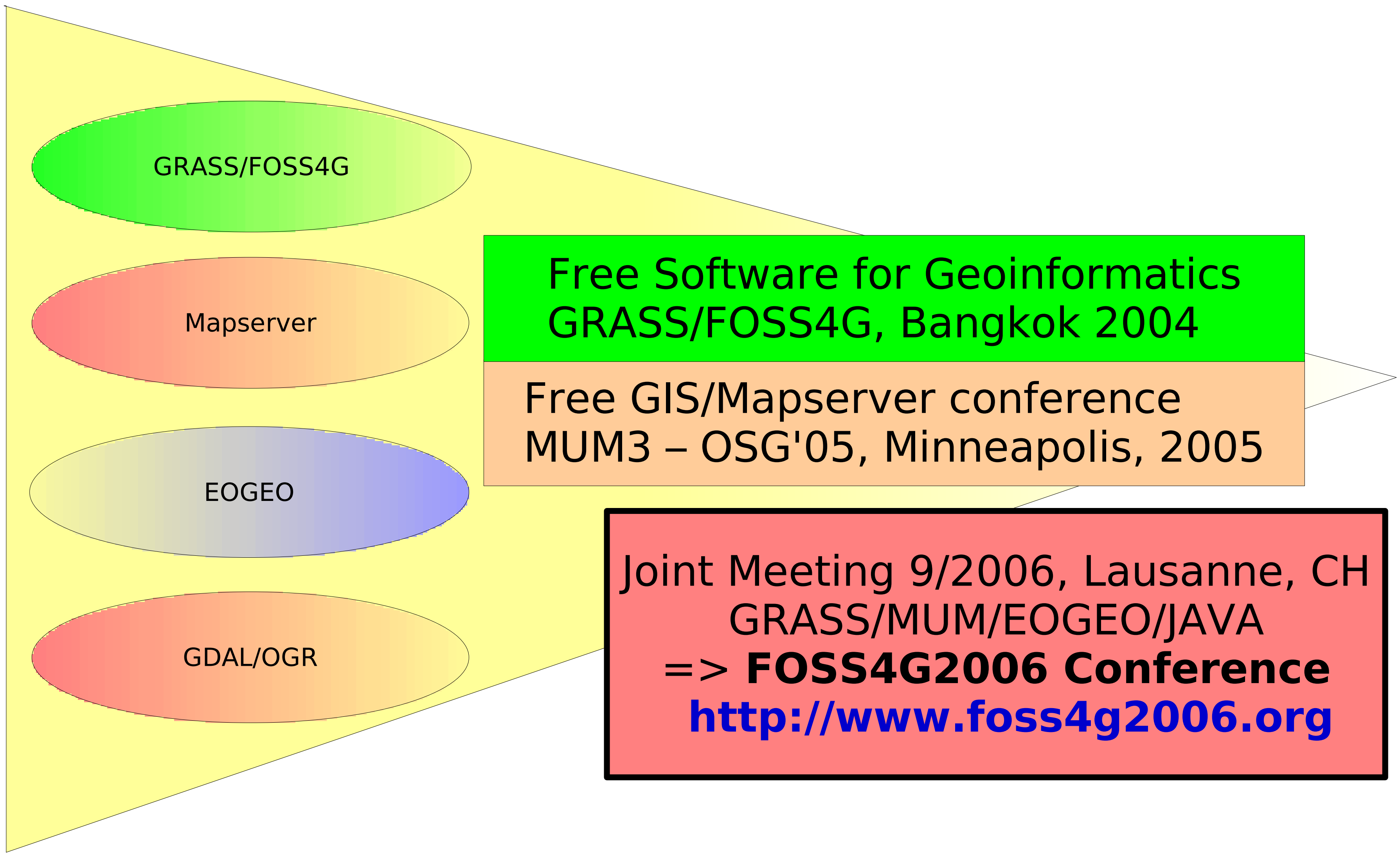
New OSGeo Foundation: Proposed founding projects

Founded 4th February 2006, Chicago

http://www.osgeo.org

Capacity building

Communities growing together...



Capacity building

Communities growing together...

(General) statistical computing environment:
http://www.r-project.org/

Rgeo: spatial data analysis in R, unified classes and interfaces (e.g, RGRASS)
http://r-spatial.sourceforge.net/

PostgreSQL
Most advanced open source relational database
http://www.postgresql.org/

GRASS GIS
Spatial Computing
http://grass.itc.it

GDAL - Geospatial Data Abstraction Library
http://www.gdal.org

PostGIS: support for geographic objects to the PostgreSQL object-relational database
http://postgis.refractions.net

QGIS: user friendly Open Source GIS
http://www.qgis.org

MAPSERVER

Spatially-enabled Internet applications
http://mapserver.gis.umn.edu/

... AND MANY OTHERS!
http://www.osgeo.org

Closure of the Seminar

Thanks for your interest and your attention!

M. Neteler
neteler at itc it
http://mpa.itc.it
http://grass.itc.it

ITC-irst, Povo (Trento), Italy

License of this document