

GRASS GIS 7 Capabilities

An overview

Markus Neteler

<http://gis.cri.fmach.it/neteler/>
<http://courses.neteler.org/blog>
<http://consulting.neteler.org/>

2015

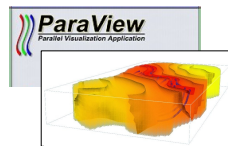


What's GRASS GIS?

<http://grass.osgeo.org>



- **Geographic Resources Analysis Support System**
- **Open Source GIS**, developed since 1984, since 1999 GNU GPL
- **Portable code** (many operating systems, 32/64bit)
- Your **GIS backbone** – linkable to:



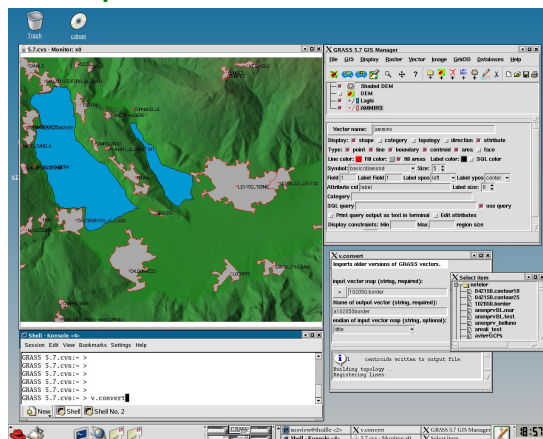
ZOO
<http://zoo-project.org>



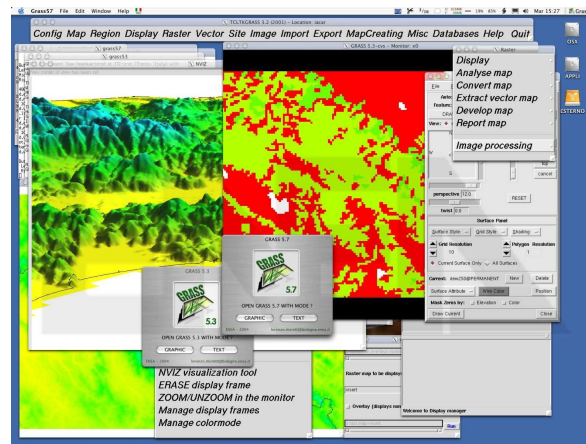
SEXTANTE



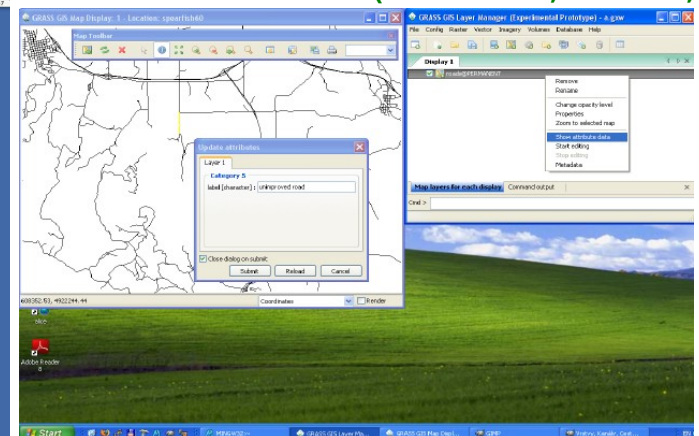
GNU/Linux



MacOSX



MS-Windows

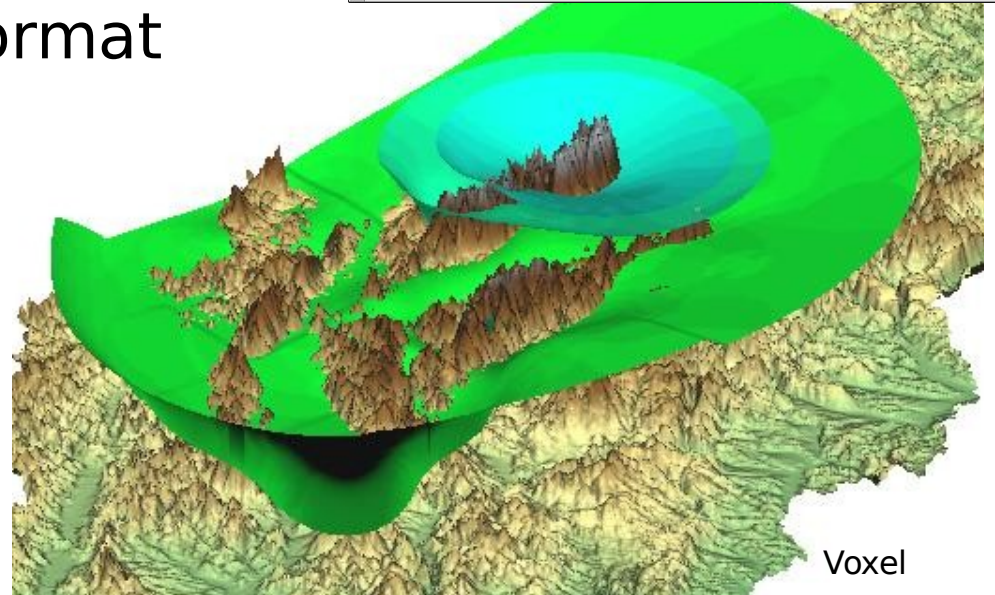
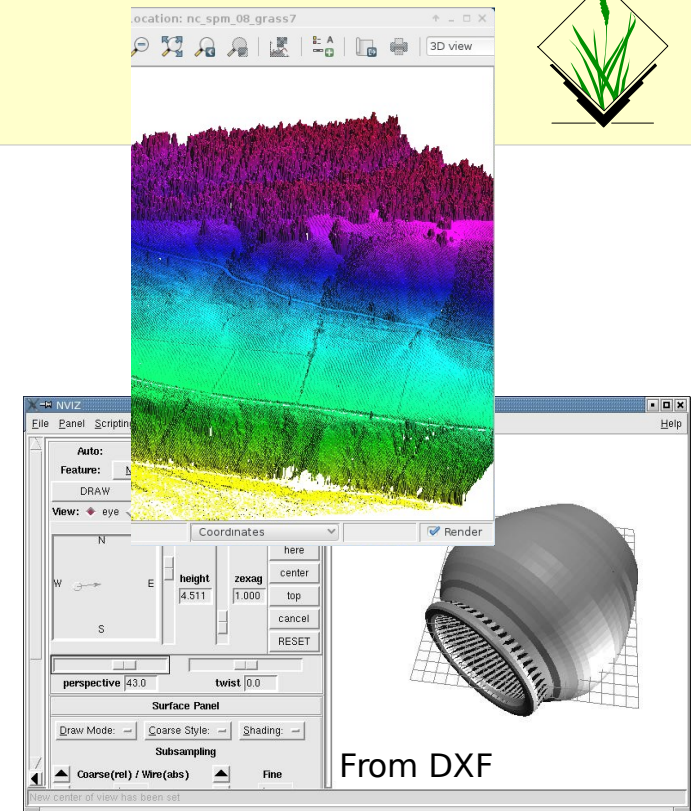


(IBM AIX, *BSD, ...)

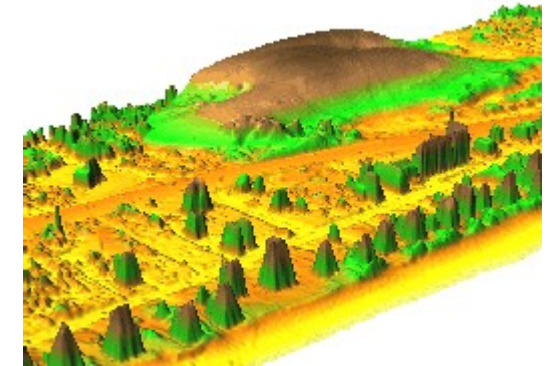
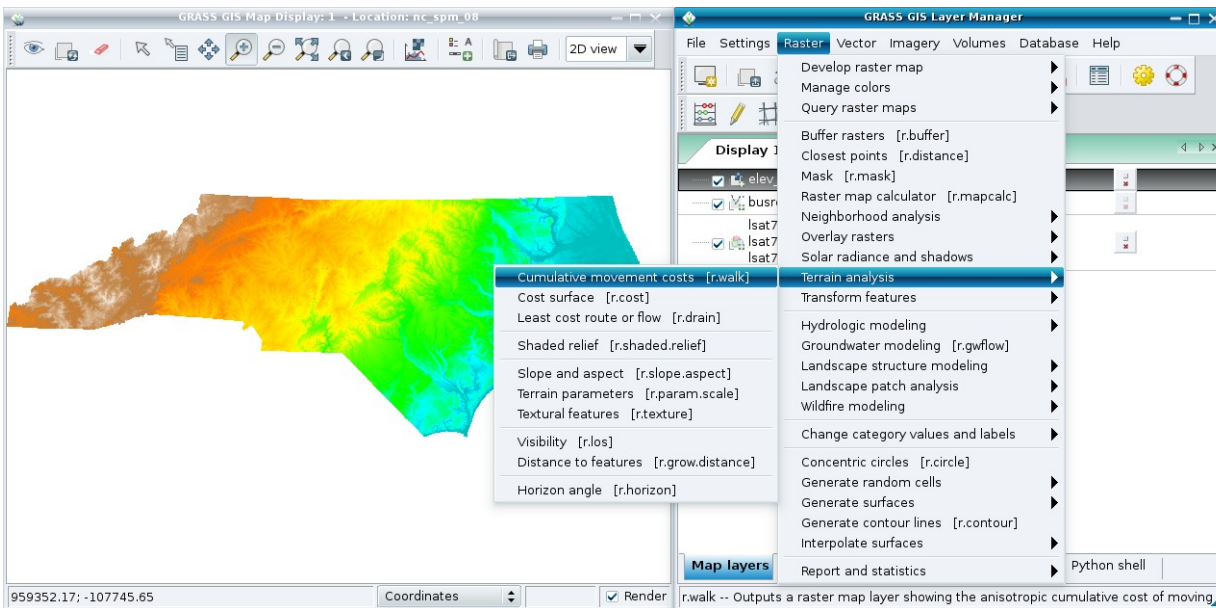
What's GRASS GIS?



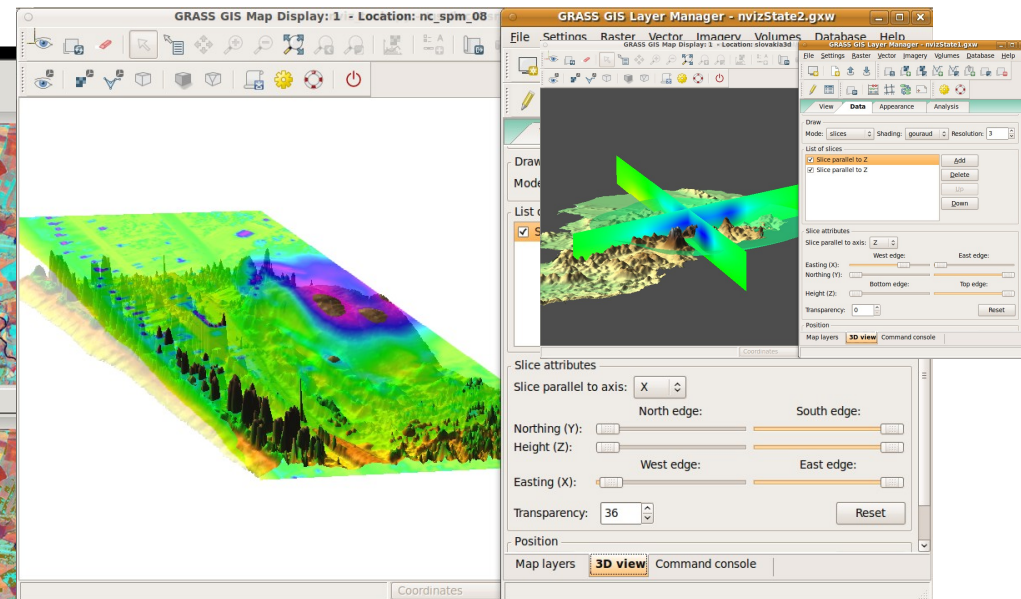
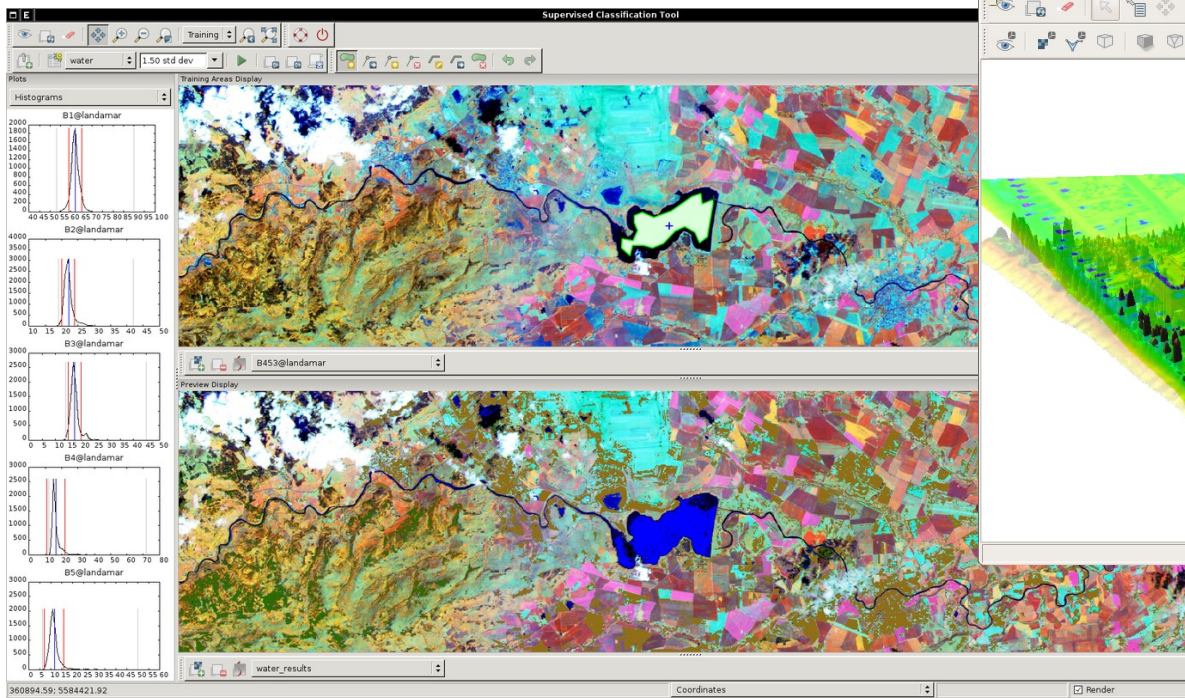
- Raster 2D/3D (voxel) processing
- Vector 2D/3D topological processing
- Vector network analysis support
- Image processing system
- Space-time cubes, temporal GIS
- Native raster and vector format
- 3D Visualization system
- DBMS integrated (SQL) with SQLite, DBF, PostgreSQL, MySQL, and ODBC drivers



GRASS GIS 7 User interface



Nagshead LiDAR time series:
dune moving over 9 years
(NC, USA) – animation



GRASS GIS 7 graphical user interface



GRASS GIS Layer Manager

GRASS GIS Map Display: 1 - Location: nc_spm_08

File Settings Raster Vector Imagery Volumes Database Help

Display 1

GRASS GIS Attribute Table Manager - <soils_wake@PERMANENT>

1 / Table soils_wake

Attribute data - right-click to edit/manage records

cat	AREA	PERIMETER	SOILS_	SOILS_ID	DSL_NAI HYDRIC
1	34308200	715717	411	45556	w
2	24420300	141501	19488	18683	CeB2
3	7864140	77984.4	20036	46346	CeC2
4	136123000	494143	20179	46348	CeB2
5	4255490	36089.4	20209	19401	CeD
6	4528200	29120.8	20246	46349	CeD
7	17014600	152098	20256	19447	CeC2
8	3248240	30345	20357	19548	CeD
9	4342250	28882.4	20833	46355	ApC2
10	1885750	22272.4	20966	20149	Cm B
11	13345800	115666	20970	20153	CeC2
12	1279590	17500.7	21030	20212	Cm D

SQL Query

Simple SELECT * FROM soils_wake WHERE AREA

Advanced SELECT * FROM soils_wake

Browse data Manage tables Manage layers

Refresh Quit

Number of loaded records: 3378

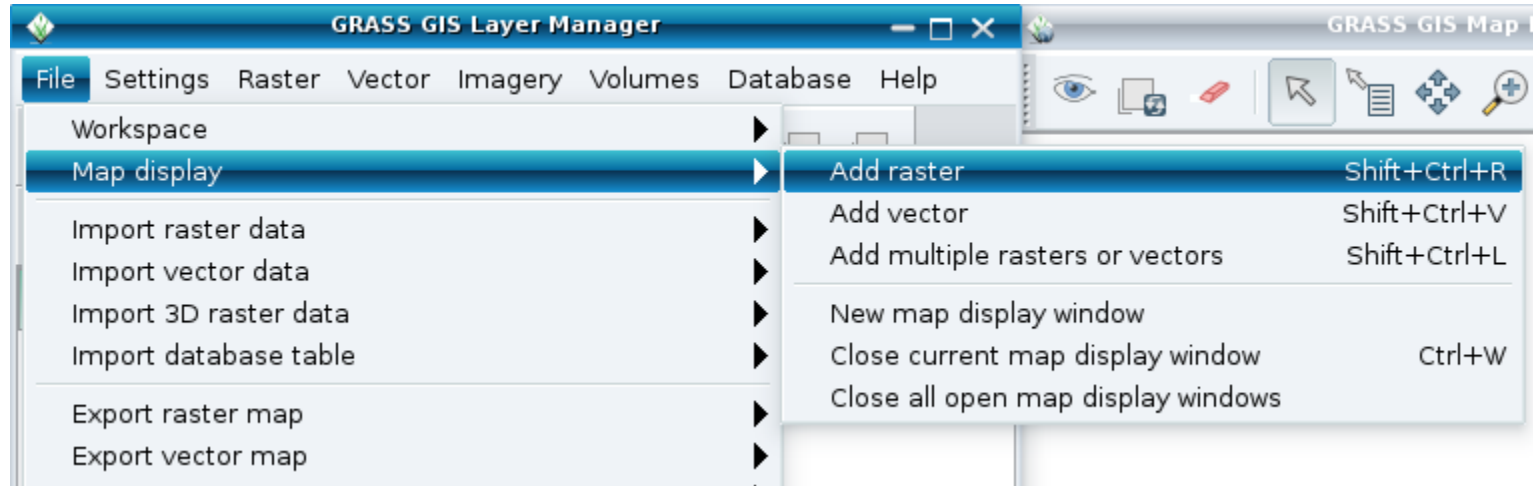
Map layers Command console Search module Python shell

d.vect -c map=soils_wake@PERMANENT 634177.17; 225212.69 Coordinates Render



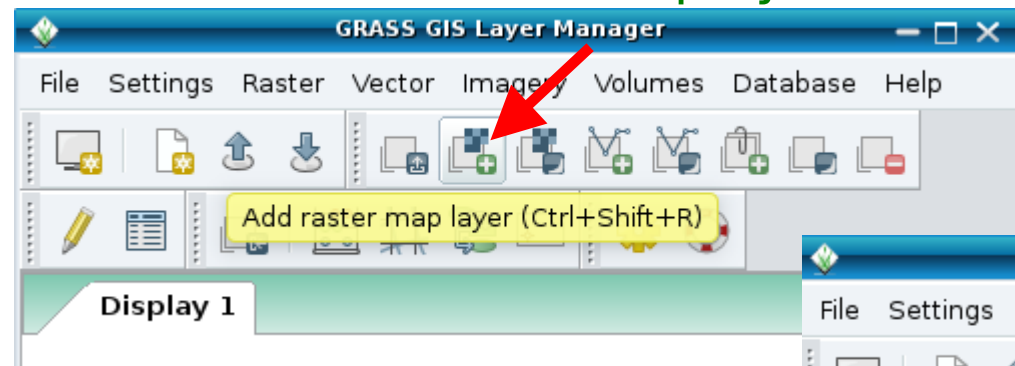
Displaying raster and vector maps

A) Using the menu

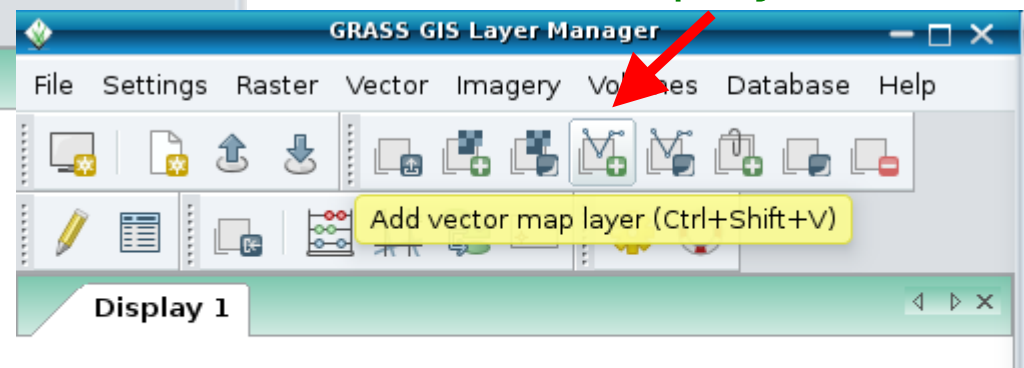


B) Using the icons

Display raster maps



Display vector maps





Showing vector map attributes

Opening the attribute table of the “roadsmajor” vector map by

- ... either *right-mouse* clicking in layer tree on map name
- ... or using the related “Show Attribute table” icon

The screenshot shows two windows from the GRASS GIS interface. The top window is the 'GRASS GIS Layer Manager' with a red arrow pointing to the 'Show attribute data for selected vector map' icon in the toolbar. The bottom window is the 'GRASS GIS Attribute Table Manager - READONLY - <roadsmajor@PERMANENT>' showing the attribute table for the 'roadsmajor' vector map.

cat	MAJORRDS_	ROAD_NAME	MULTILA	PROPYEAR	OBJECTID	SHAPE_LEN
1	1	NC-50	no	0	1	4825.369405
2	2	NC-50	no	0	2	14392.589058
3	3	NC-98	no	0	3	3212.981242
4	4	NC-50	no	0	4	13391.907552
5	5	NC-98	no	0	5	7196.001495
6	6		no	0	6	10185.513951
7	7	US-1	yes	0	7	13655.438596
8	8		no	0	8	797.901095
9	9	NC-98	no	0	9	14772.176241
10	10	NC-98	no	0	10	8446.822876
11	11	NC-98	no	0	11	14876.323626
12	12	NC-98	no	0	12	11610.268716

SQL Query: Advanced |
SELECT * FROM roadsmajor WHERE cat = [] Apply

Browse data | Manage tables | Manage layers

Refresh Close

Number of loaded records: 355



SQL queries of attributes

Selecting the **single lane roads** in the “roadsmajor” vector map

- Use “Simple” SQL query
- The selected vectors will be highlighted in the map display

The screenshot shows the GRASS GIS Attribute Table Manager window for the 'roadsmajor' table. The table contains the following data:

cat	MAJORRDS_	ROAD_NAME	MULTILANE	PROPYEAI	OBJECTID	SHAPE_LEN
1	1	NC-50	no	0	1	4825.369405
2	2	NC-50	no	0	2	14392.589058
3	3	NC-98	no	0	3	3212.981242
4	4	NC-50	no	0	4	13391.907552
5	5	NC-98	no	0	5	7196.001495
6	6		no	0	6	10185.513951
8	8					
9	9	NC-98				
10	10	NC-98				
11	11	NC-98				
12	12	NC-98				

The SQL Query dialog box is open, showing the following query:

```
SELECT * FROM roadsmajor WHERE MULTILANE = 'no'
```

The dialog box also includes a 'Simple' radio button selected, an 'Advanced' radio button, and a 'SQL Builder' button. The 'Number of loaded records' is 115.



Adding map elements

Using the Wake county “elevation” and “roadsmajor” maps:

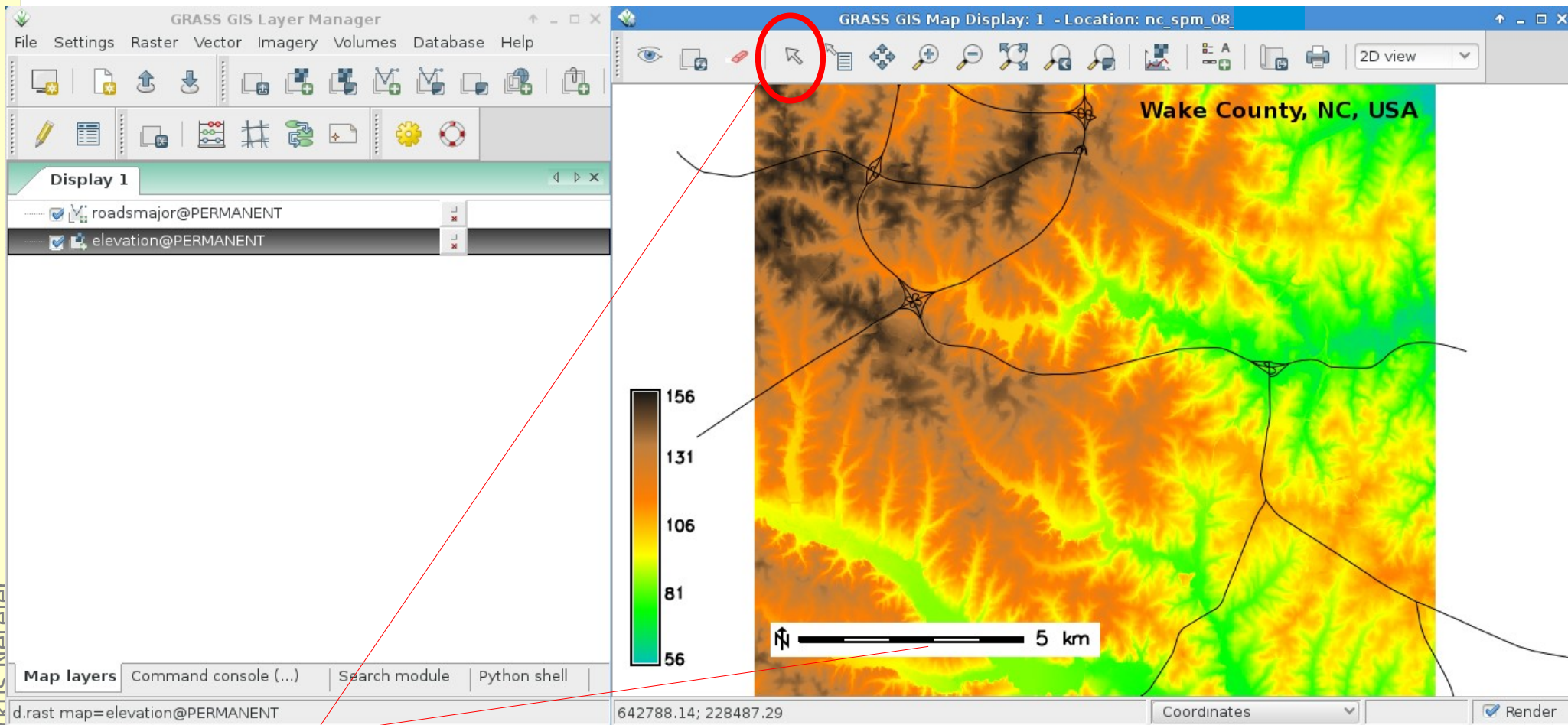
- ☰ Add scalebar and north arrow
- ☰ Add legend
- ☰ Add text layer

The screenshot shows the GRASS GIS interface. On the left is the 'GRASS GIS Layer Manager' window with a red arrow pointing to the 'Add' button. The main window is 'GRASS GIS Map Display: 1 - Location: nc_spm_0'. It displays a map of Wake County, NC, USA, with an elevation raster and major roads. A legend is visible on the left side of the map, showing a color scale from 56 to 156. A scale bar and north arrow are also present. The bottom status bar shows the command 'd.rast map=elevation@PERMANENT' and coordinates '642788.14; 228487.29'.



Modifying element settings and position

Using the Wake county “elevation” and “roadsmajor” maps:

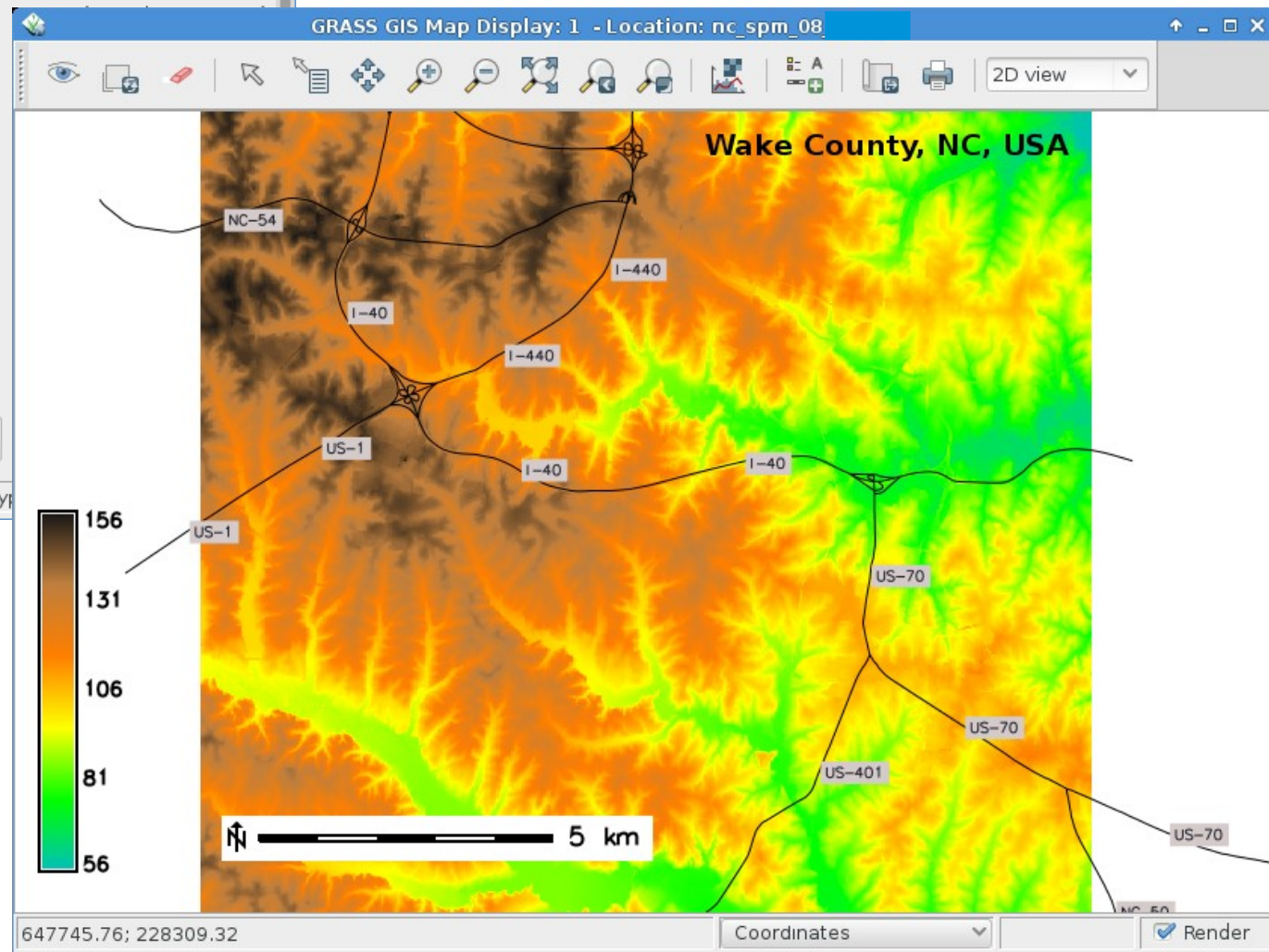
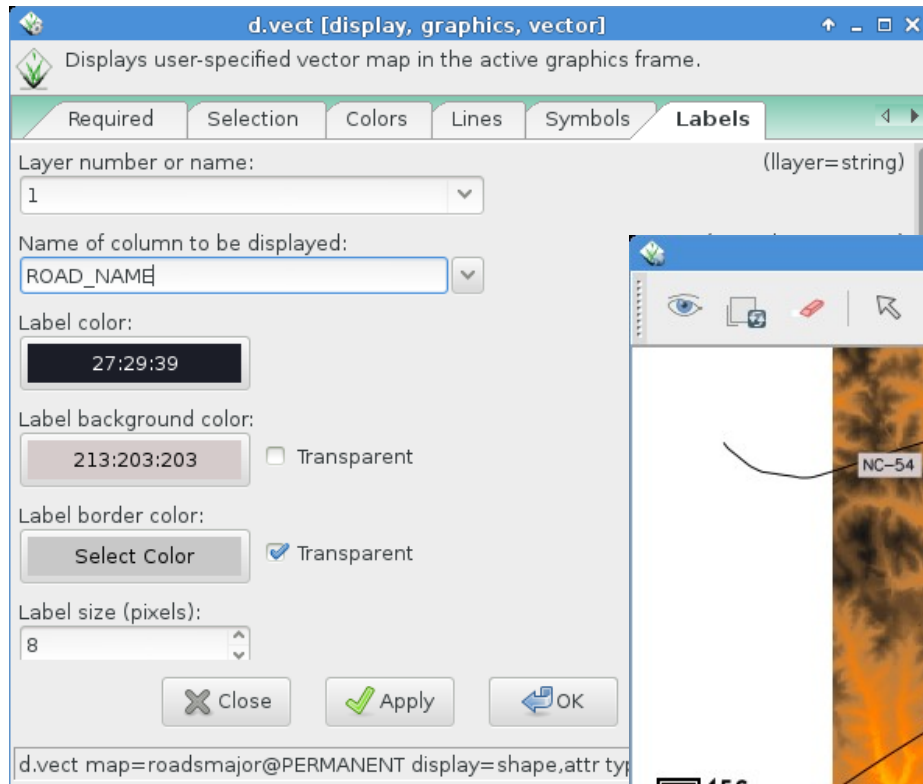


- Use pointer to
- move map elements
 - edit element settings with a click



Adding vector map labels

Activating vector labels for “roadsmajor” map: right-click map in Layer Manager, Properties



For a more sophisticated labeling system, see [v.label](#) and [v.label.sa](#).



Map histogram tool

Using the Wake county “elevation” map:

GRASS GIS Map Display: 1 - Location: nc...m_08

- Measure distance
- Profile surface map
- Create bivariate scatterplot of raster maps
- Create histogram of raster map**
- Create histogram with d.histogram
- Vector network analysis tool

Select raster map or imagery group to histogram

Histogram single raster Histogram imagery group

Select raster map: elevation@PERMANENT

Select image group:

Number of bins (for FP maps) 255

Histogram type count

Cancel OK

GRASS Histogramming Tool

Histogram of elevation

Cell counts

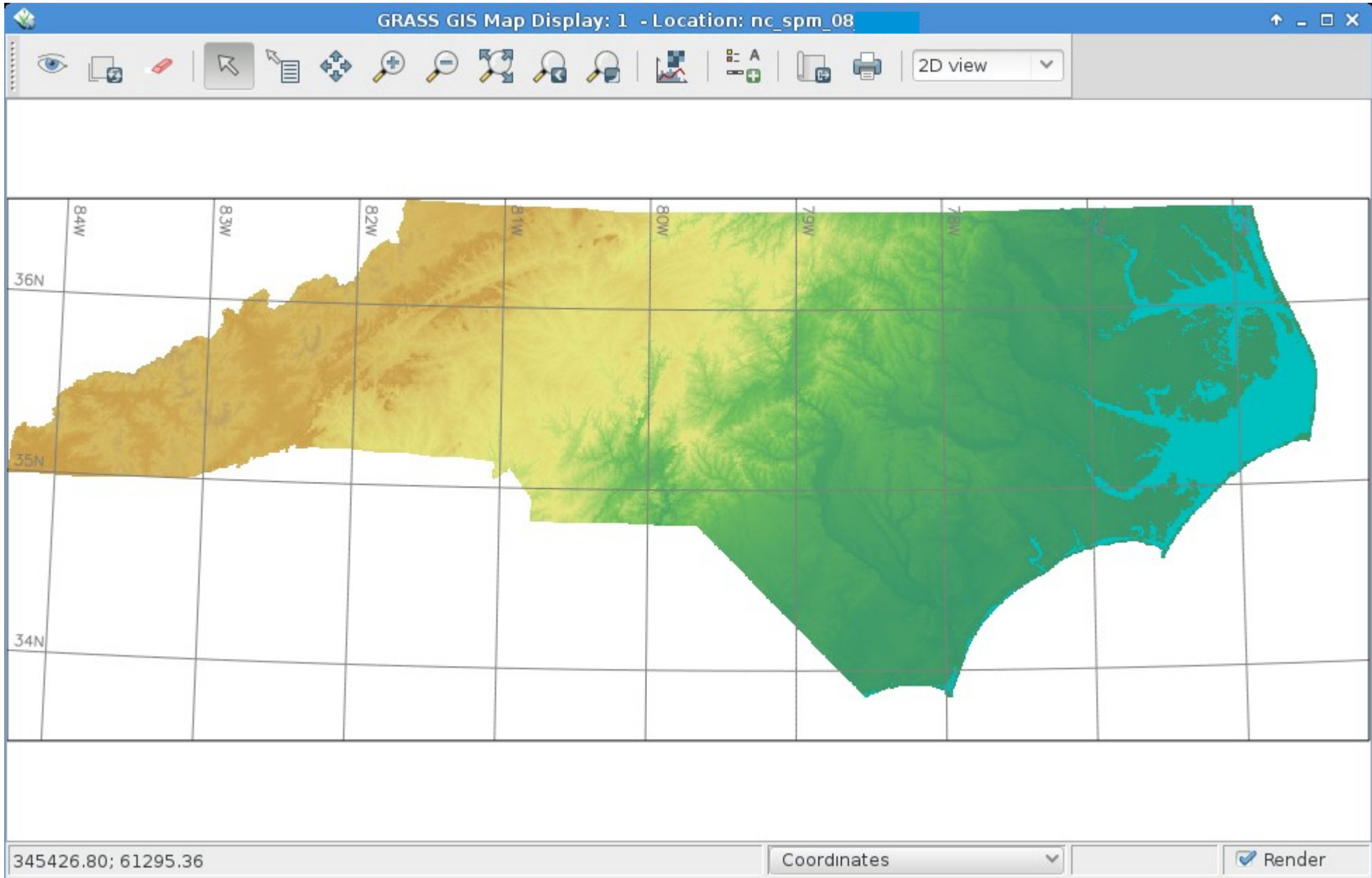
Raster cell values

Map will be preselected if selected in Layer Manager



Adding a grid to the map

Using the NC state “elev_state_500m” map:



GRASS GIS 7 Vector features

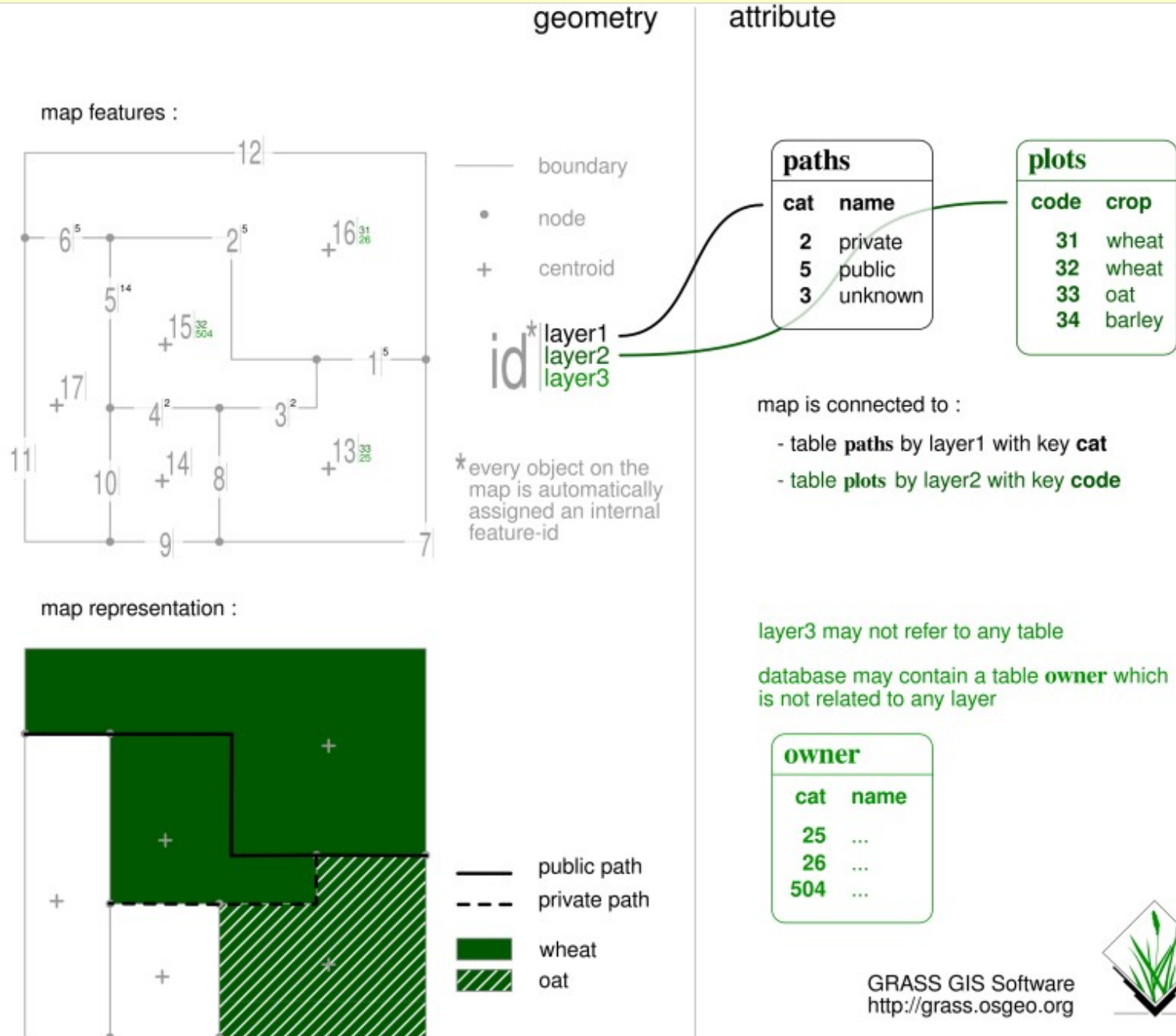


Native vector format

- Vector topology
- m:n mapping of geometry features to attributes
- Support of vector layers
- OGC Simple Features \leftrightarrow Topological Vector Conversion
- Database Management system (DBMS) with SQL support
- SQLite (default DB backend), PostgreSQL + PostGIS, MySQL, ODBC (, DBF)

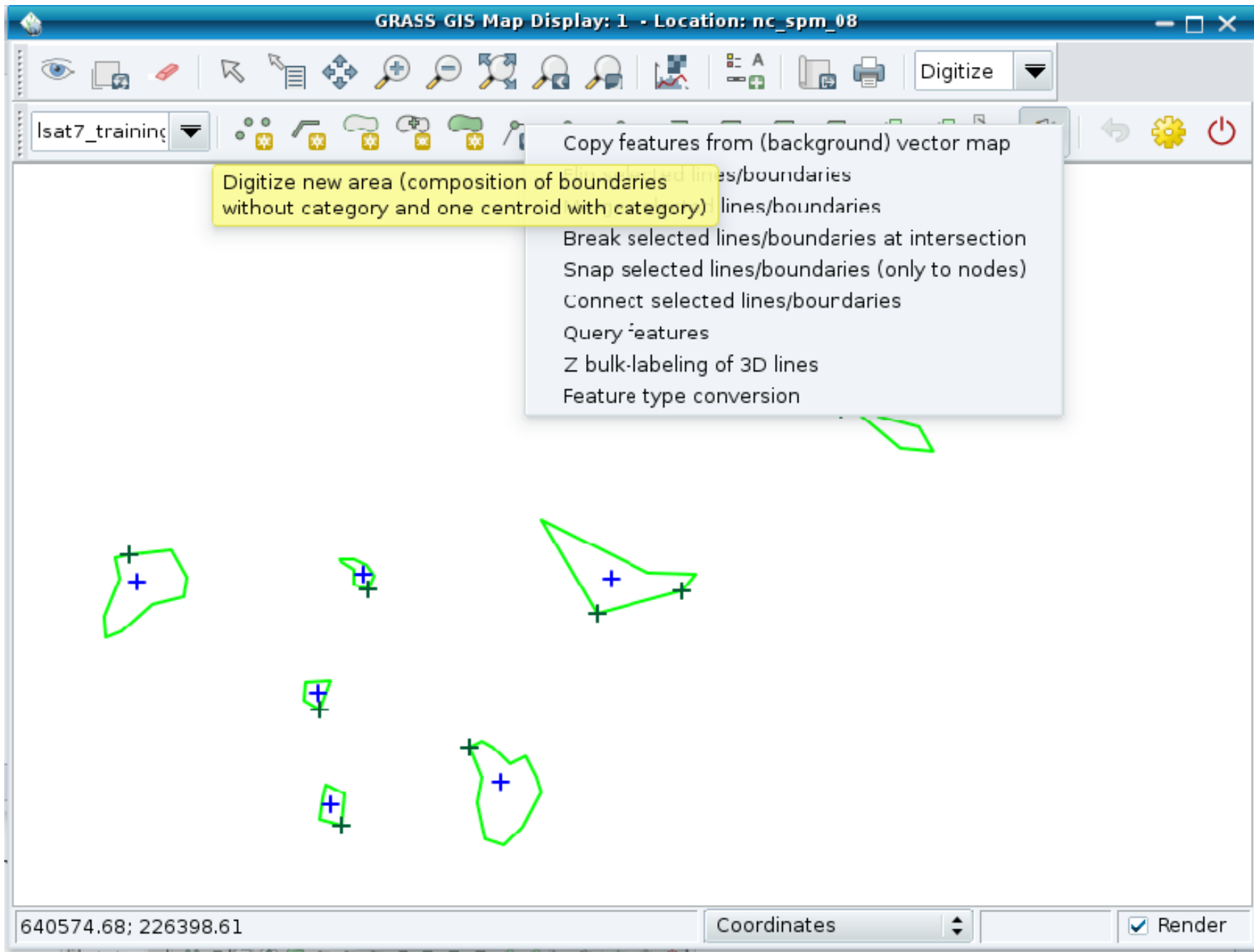


GRASS GIS 7 Vector features





GRASS GIS topological vector digitizer



Further vector processing capabilities



Example vector module groups

Topological geometry feature digitizing/editing

<https://grass.osgeo.org/grass70/manuals/vectorintro.html>

LiDAR analysis:

<http://grasswiki.osgeo.org/wiki/LIDAR>

Linear referencing (LRS) – v.lrs.*:

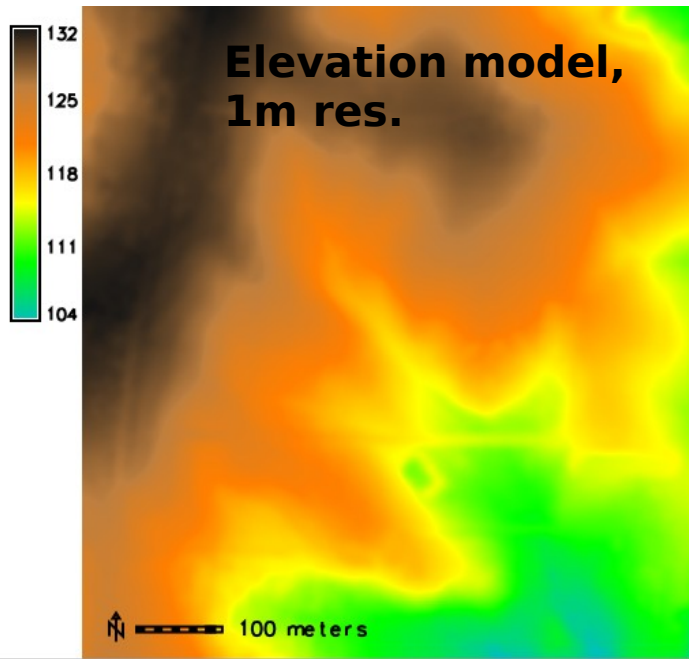
http://grasswiki.osgeo.org/wiki/Linear_Reference_System

Network analysis – v.net.*:

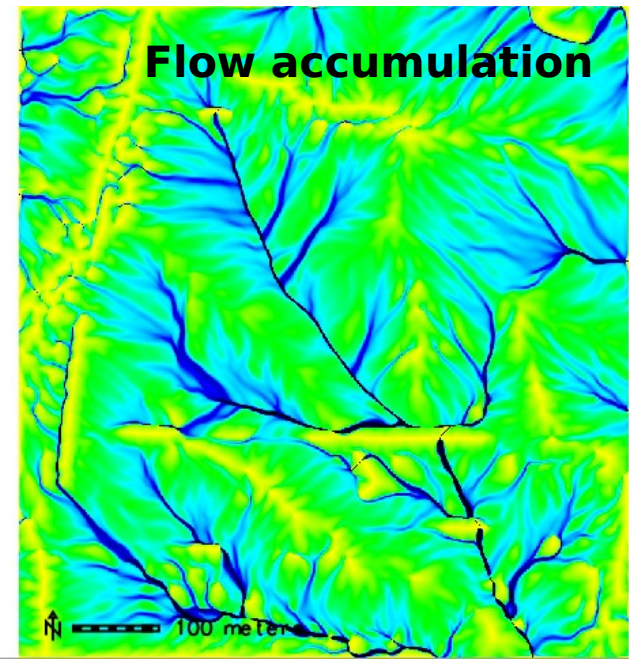
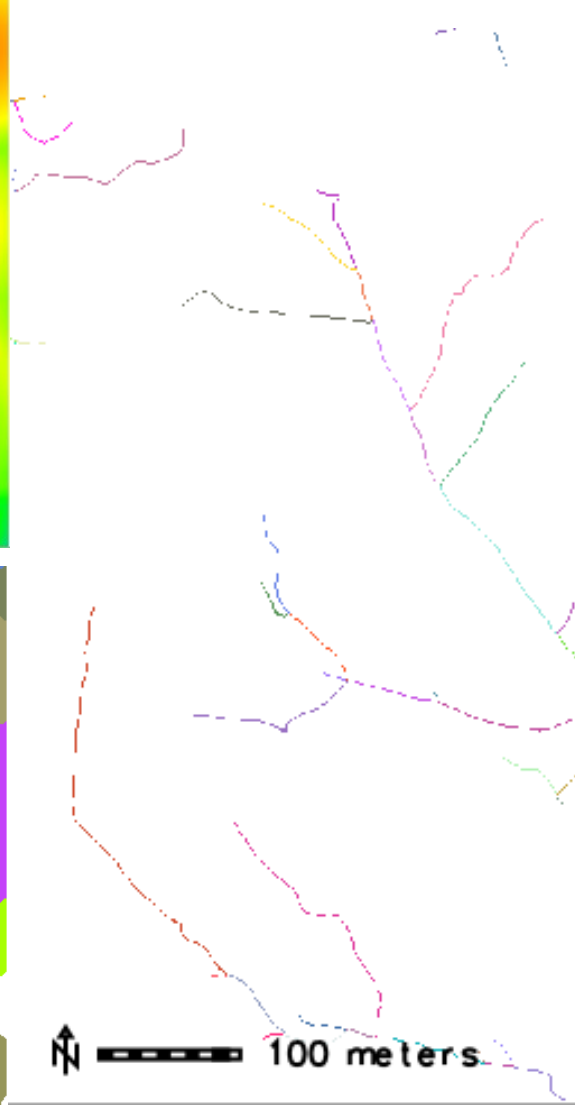
http://grasswiki.osgeo.org/wiki/Vector_network_analysis



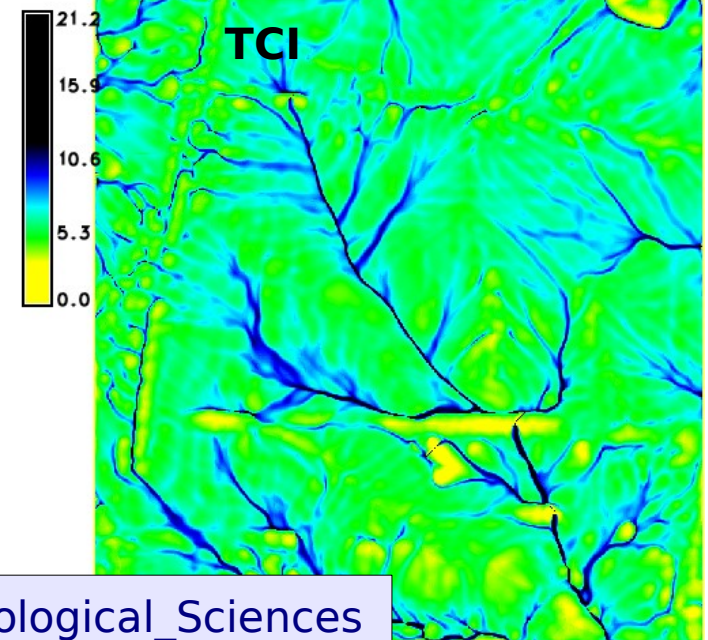
Hydrological analysis: examples



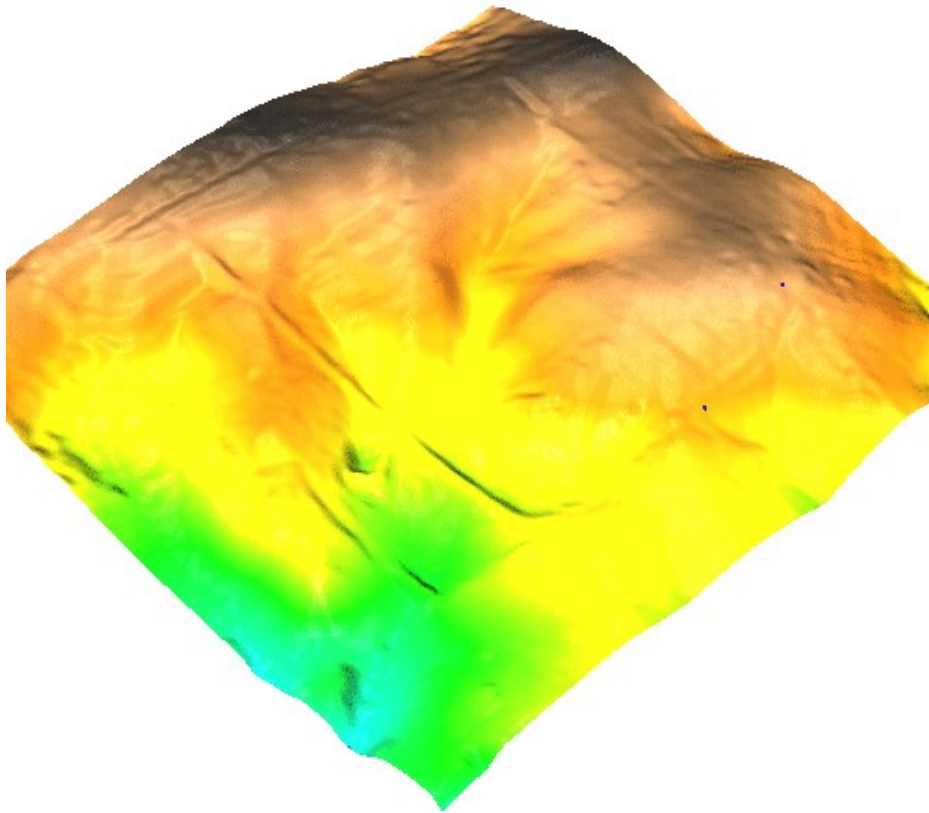
Stream network



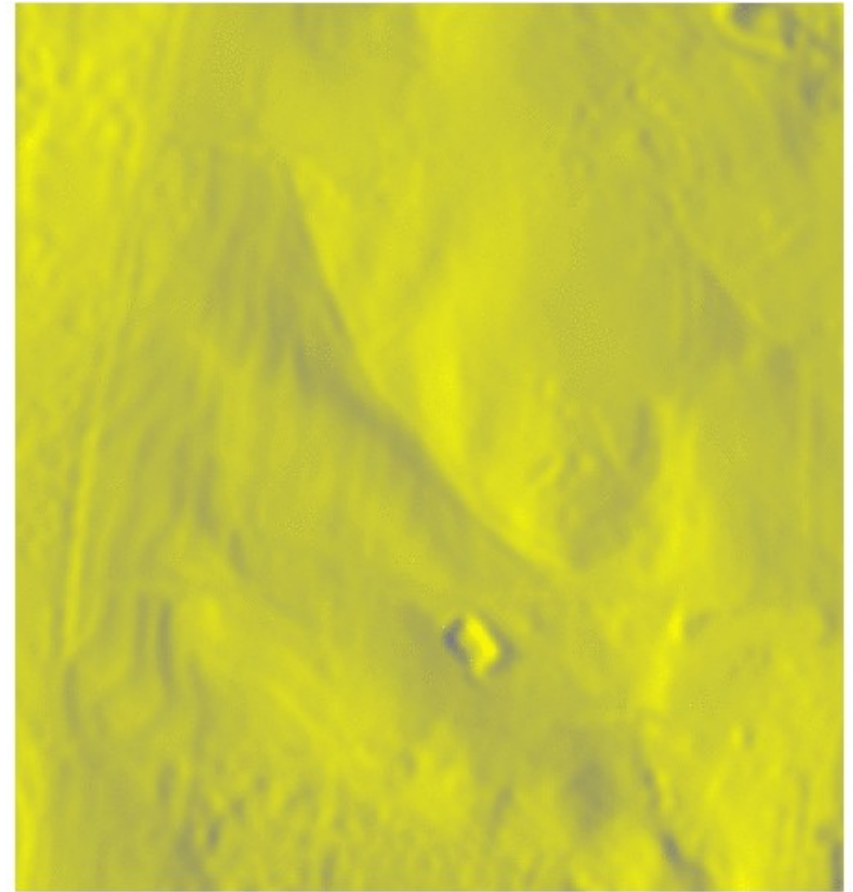
multiple flow direction



Hydrological analysis: examples



Particle flow over time
(animation)

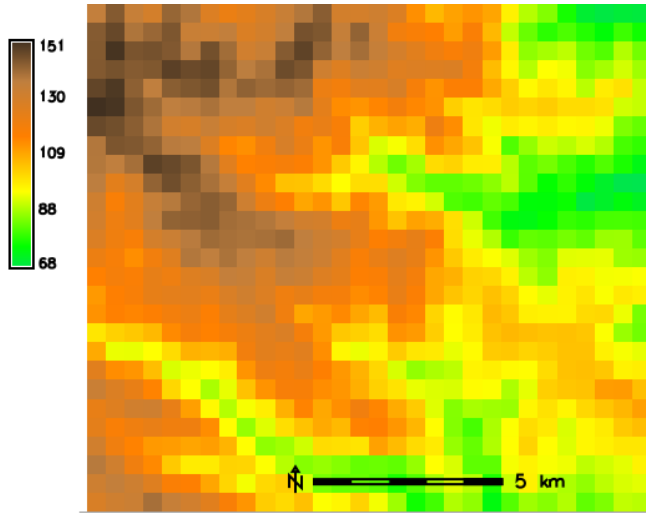


Accumulation over time
(animation)

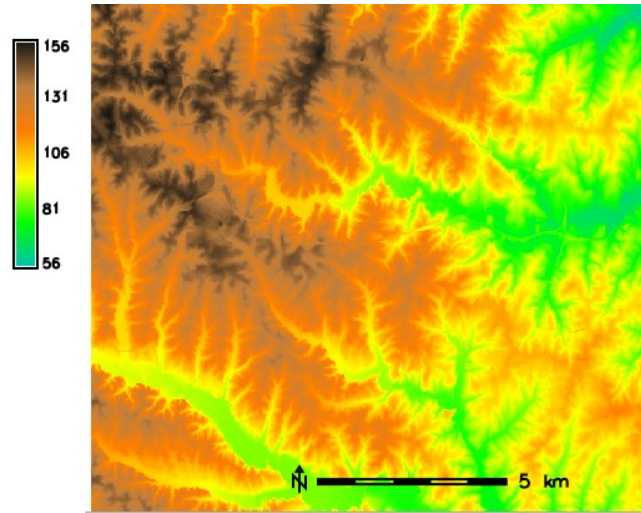
Credit:
Helena Mitsova, NCSU ([web](#))



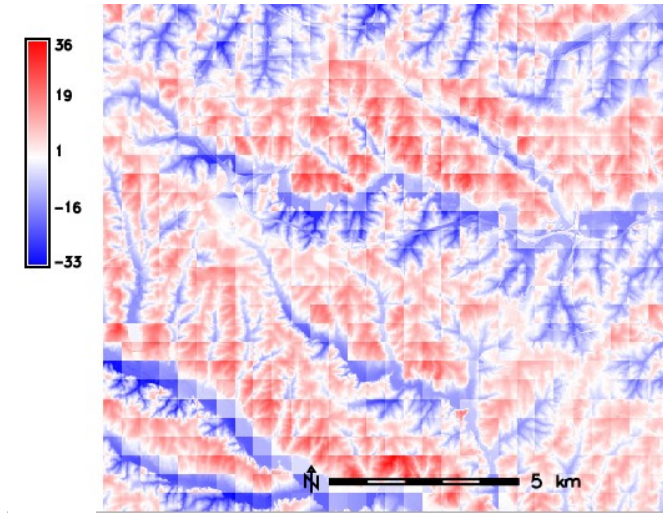
Raster resampling with interpolation



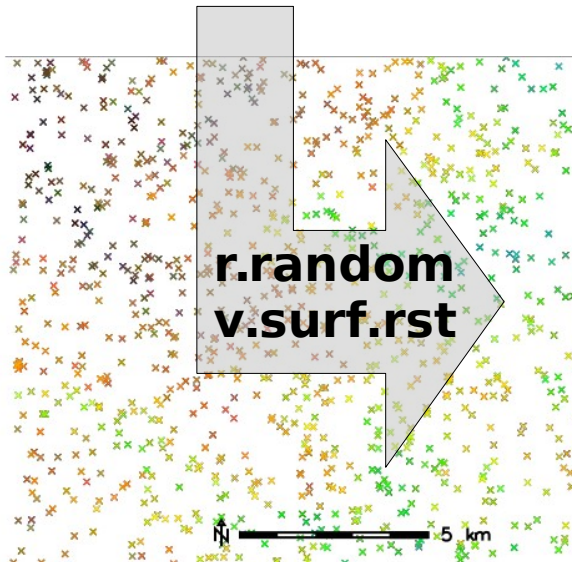
elev_state_500m (orig)



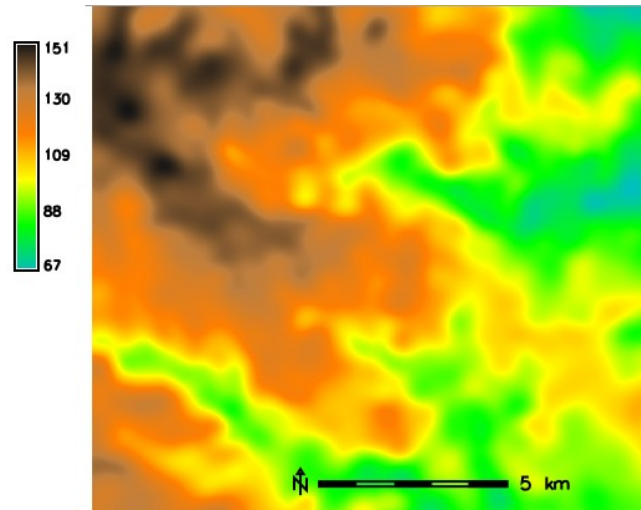
elevation 10m (orig)



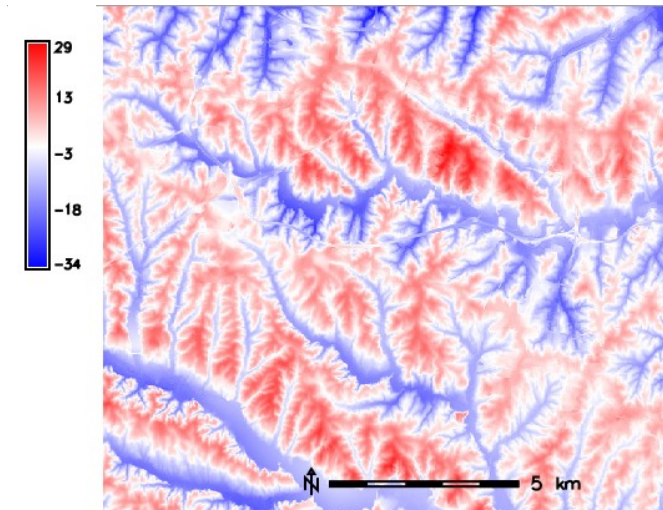
differences at 10m



920 random points
(optionally: add other sources)



RST: elev state at 10m



differences at 10m



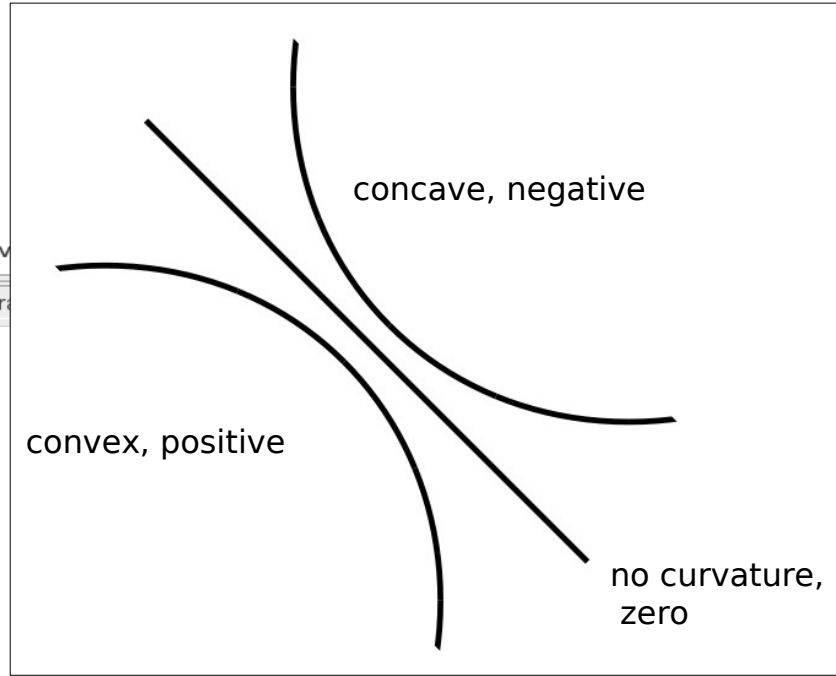
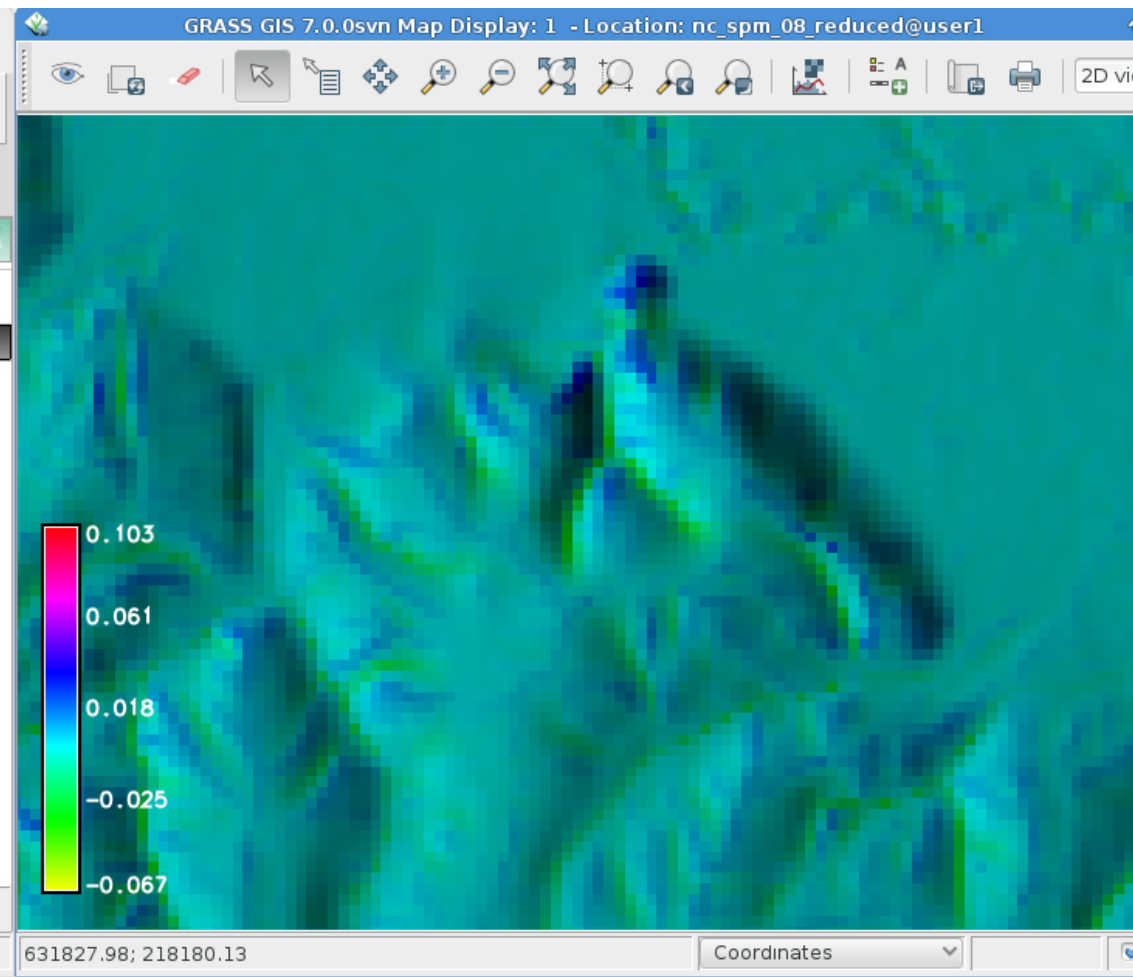
Calculation of terrain curvatures

GRASS GIS 7.0.0svn Layer Manager

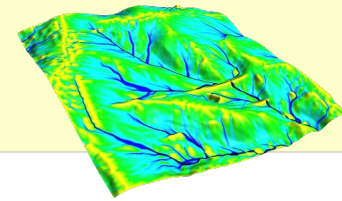
File Settings Raster Vector Imagery Volumes Database Temporal Help

Display 1

- elevation.tcurf3@user1
- elevation.shade@user1
- elevation.tcurf3@user1



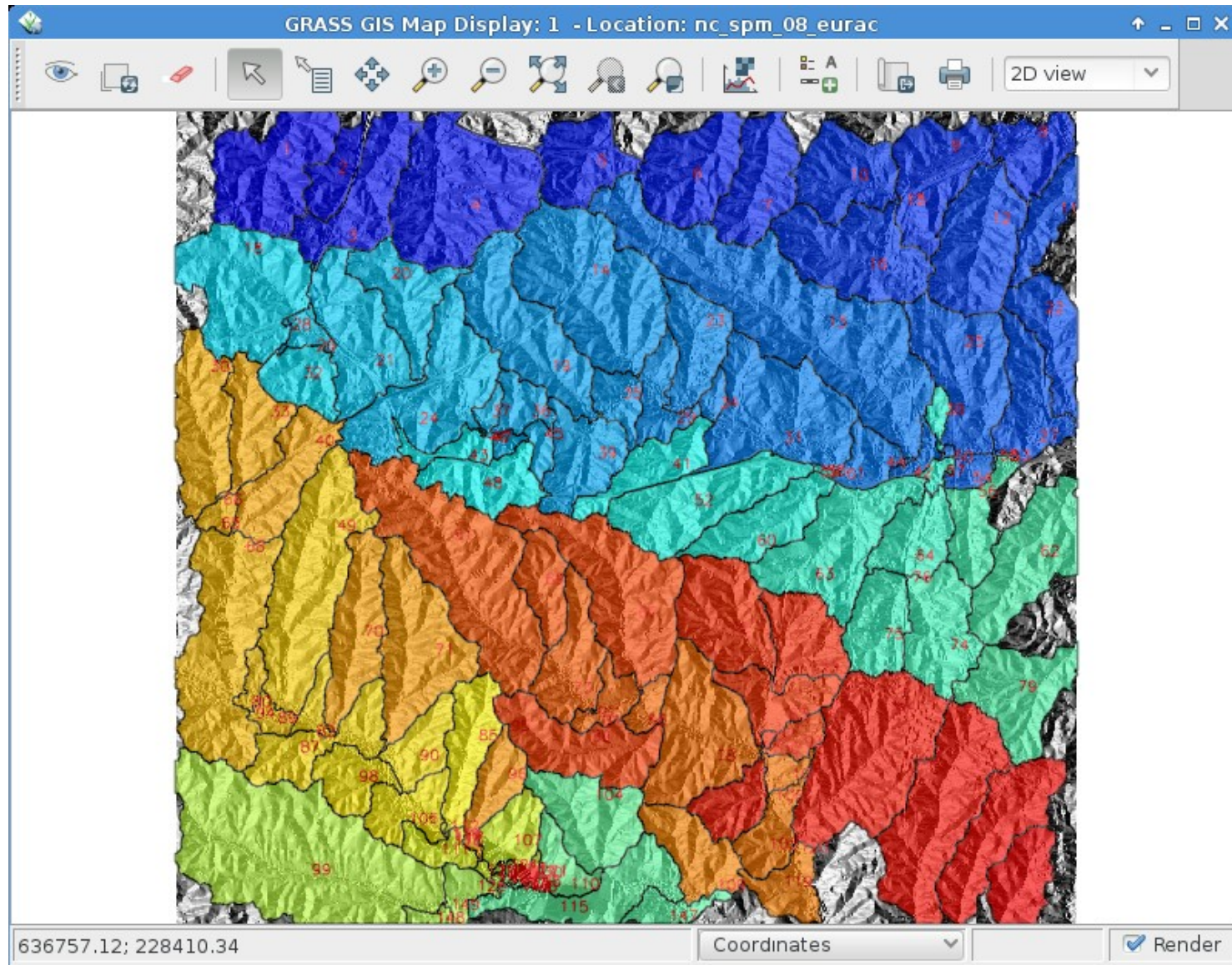
Raster data analysis: further methods



- 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**, **r.walk**
- Energy:
 - cast shadows, astronomical calculations of sun position: **r.sunmask**
 - energy budget: **r.sun**
- Line of sight:
 - viewsheds can be generated with: **r.viewshed**
- Interpolation methods
 - 2D inverse distance weighted: **v.surf.idw**
 - 2D from contour lines: **r.surf.contour**
 - 2D bilinear: **r.resamp.interp**
 - 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**

... and much more!

Raster-vector statistics: DEM stats per watershed



With **v.rast.stats** we will calculate univariate statistics per polygon based on a raster map.

Results are added as new columns to the attribute table of the watershed map.

Raster-vector statistics: DEM stats per watershed



Zonal statistics of elevation data per watershed basin

```
v.rast.stats map=basins_10k raster=elevation \  
  column_prefix=elev \  
v.db.select basins_10k separator=comma
```

Statistics per watershed ID ...

GRASS GIS Attribute Table Manager - <basin_10k@user1>

1 / Table basin_10k

Attribute data - right-click to edit/manage records

cat	basin_num	label	elev_n	elev_min	elev_max	elev_range	elev_mean	elev_stddev	elev_variance	elev_cf_var	elev_sum
1	2		30346	111.7435...	155.73408...	43.990562...	137.80064...	9.27004685...	85.93376870...	6.72714311...	4181698.48722...
2	4		9220	118.4454...	152.26892...	33.823493...	134.87729...	8.14500692...	66.34113775...	6.03882748...	1243568.62303...
3	6		22504	102.3183...	156.09910...	53.780746...	134.18460...	12.1913137...	148.6281301...	9.08547887...	3019690.29414...
4	8		45764	102.8480...	155.17250...	52.324432...	133.37537...	10.5128660...	110.5203522...	7.88216405...	6103790.76357...
5	10		17748	100.6119...	155.16319...	54.551208...	132.90083...	11.2063305...	125.5818443...	8.43209973...	2358723.93410...
6	12		23611	90.14228...	144.26728...	54.125	120.50823...	11.2829864...	127.3057822...	9.36283435...	2845319.93252...
7	14		20817	80.41838...	133.44969...	53.031311...	113.94928...	11.1785386...	124.9597255...	9.81010003...	2372082.21596...
8	16		13383	55.99602...	90.988014...	34.991989...	71.039377...	9.09764033...	82.76705959...	12.8064753...	950719.985893...
9	18		23600	61.72412...	106.42794...	44.703815...	84.402294...	9.29577272...	86.41139048...	11.0136492...	1991894.16147...
10	20		20129	74.98153...	128.39311...	53.411575...	102.35722...	11.7841064...	138.8651646...	11.5127257...	2060348.56206...
11	26		10951	57.97982...	108.12604...	50.146217...	86.146171...	10.4951096...	110.1473270...	12.1829089...	943386.728965...
12	24		28814	61.66728...	109.61430...	47.947013...	89.160832...	10.2329887...	104.7140597...	11.4770000...	2569080.22002...

SQL Query

Simple | Advanced

SELECT * FROM basin_10k WHERE cat =

Apply

Browse data | Manage tables | Manage layers

Refresh | Close

Number of loaded records: 149

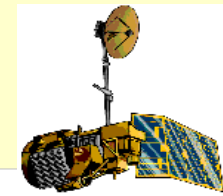


Georectification of a scanned map

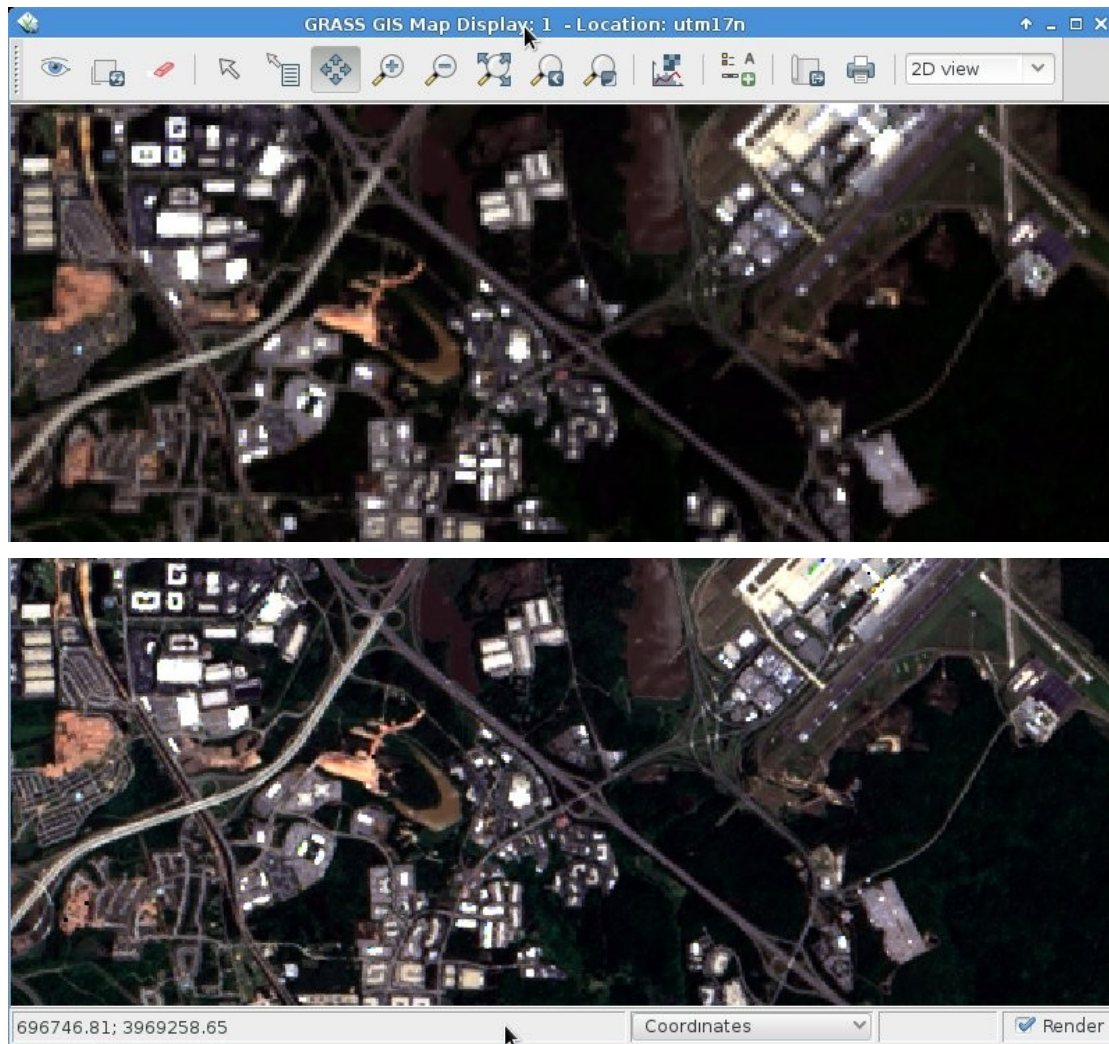
[g.gui.gcp](#): Insertion of GCP pairs (source – target):

use	source E	source N	target E	target N	Forward error	Backward error
<input type="checkbox"/> 1	835.683184107	319.142809036	641379.805188	224136.873328	0	0
<input type="checkbox"/> 2	568.387454697	168.065582194	638136.560802	226733.229199	0	0
<input type="checkbox"/> 3	568.080661387	168.372375503	638531.638625	223711.323927	0	0
<input type="checkbox"/> 4	0.0	0.0	0.0	0.0		

Image fusion – pansharpening



LANDSAT 8: Improving the R/G/B resolution with panchromatic band



R/G/B composite at 28m

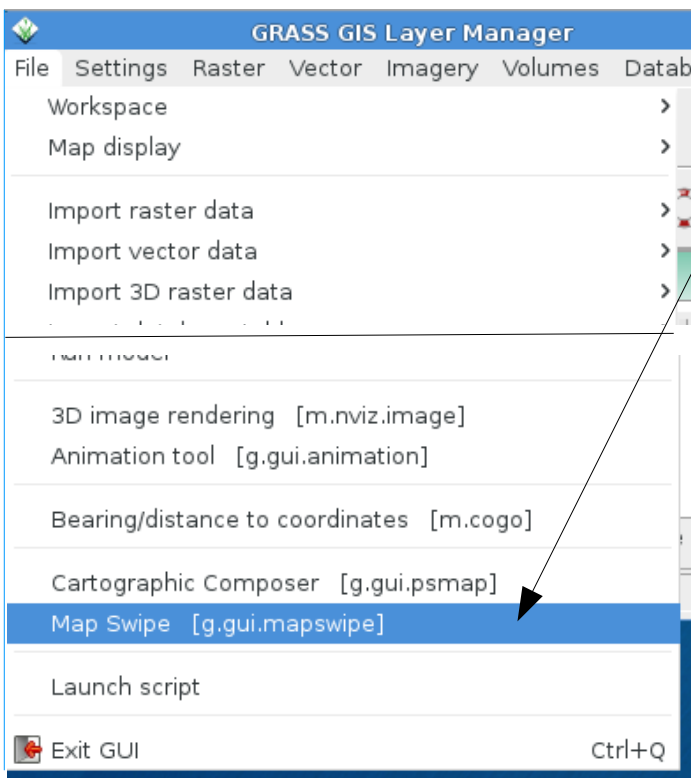
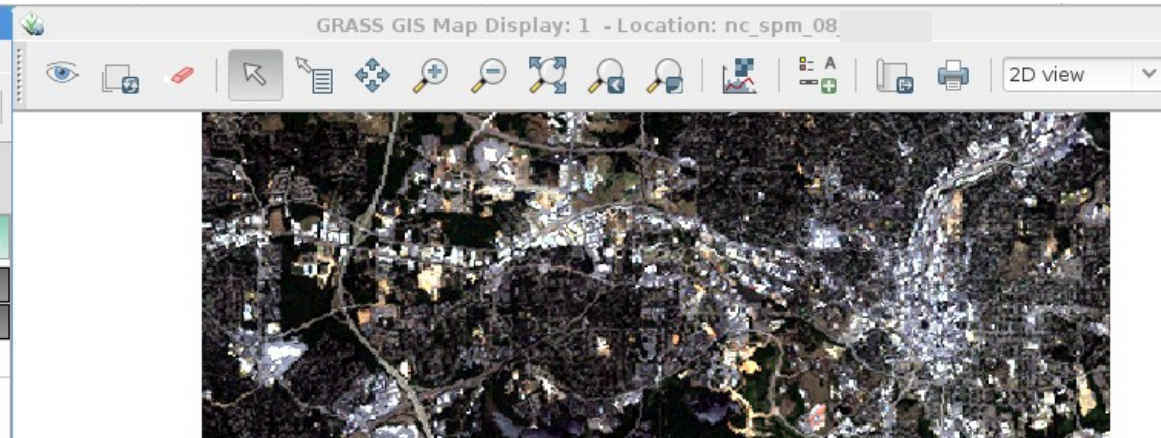
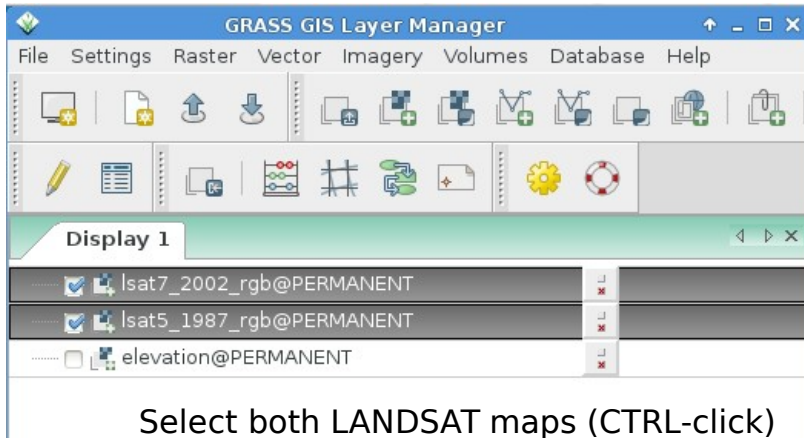
R/G/B composite at 14.25m
(IHS method, color balanced)

Note: the colors depend on the method used for pansharpening. Several methods are provided in **[i.pansharpen](#)**

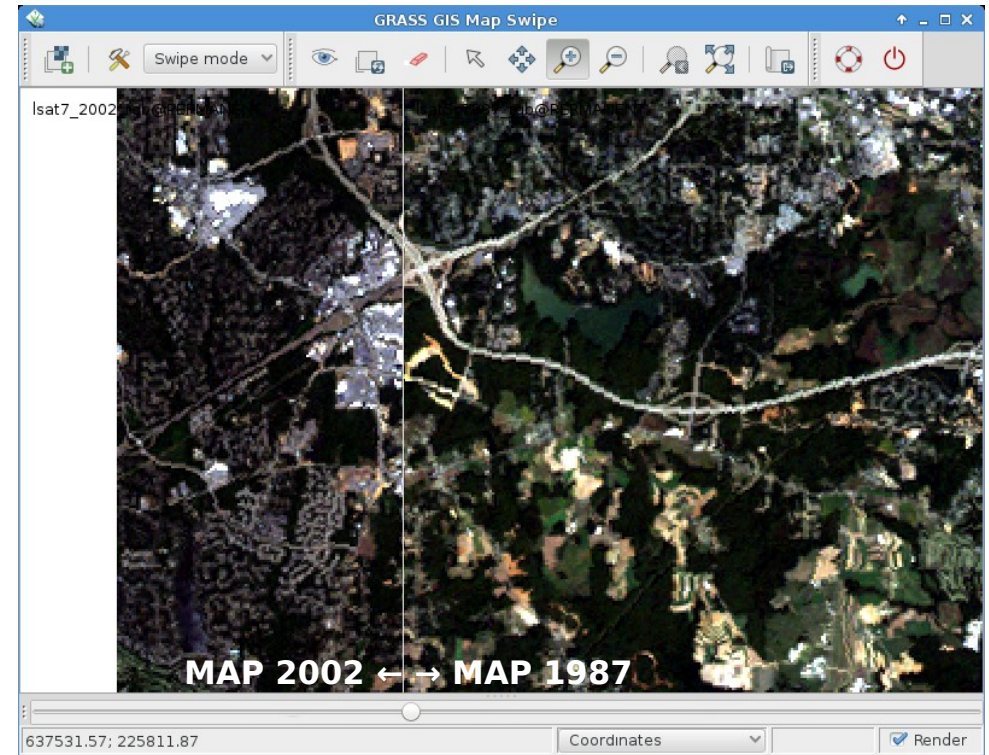


Map swiping for multitemporal maps

Comparing the LANDSAT 5 (1987) and 7 (2002) **RGB composites** of Wake county:



Switch to "map swipe" view





Bivariate Scatterplots

LANDSAT 7 2002 channels 1 and 3 of Wake county.

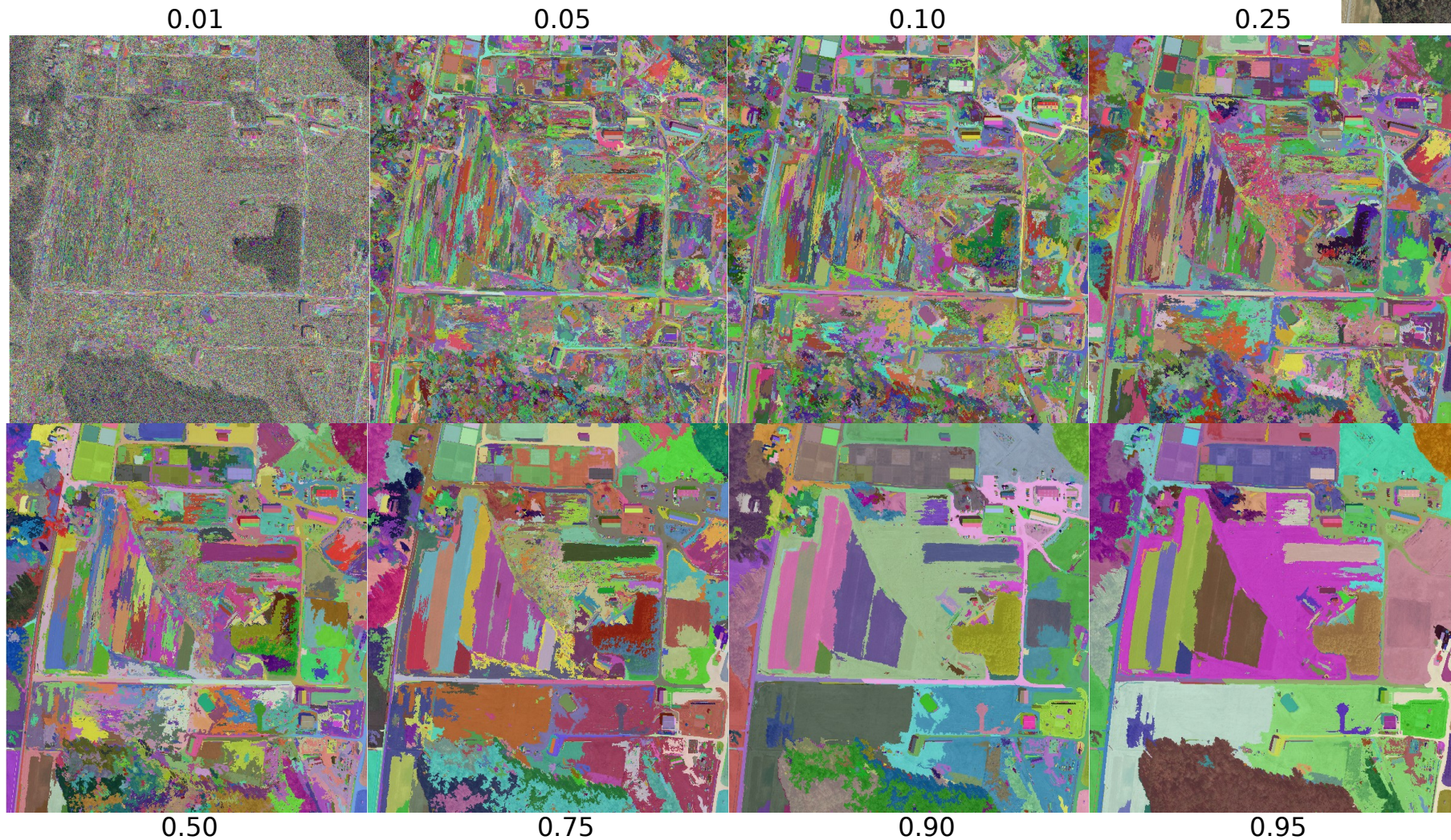
The image displays the GRASS GIS interface. On the left, the 'GRASS GIS Layer Manager' window shows three layers: 'lsat7_2002_30@PERMANENT', 'lsat7_2002_10@PERMANENT', and 'elevation@PERMANENT'. The 'Display 1' window shows a grayscale satellite image of a landscape. The 'GRASS Bivariate Scatterplot Tool' window is open, showing a scatterplot titled 'Bivariate Scatterplot of lsat7_2002_30 vs lsat7_2002_10'. The plot shows a strong positive correlation between the two channels, with both axes ranging from 0 to 300. A legend in the top right of the plot area identifies the data as 'lsat7_2002_30 vs lsat7_2002_10'. A context menu is visible over the map display, with 'Create bivariate scatterplot of raster' selected.

Unsupervised image classification – Segmentation



Re-use output of previous run (seed):

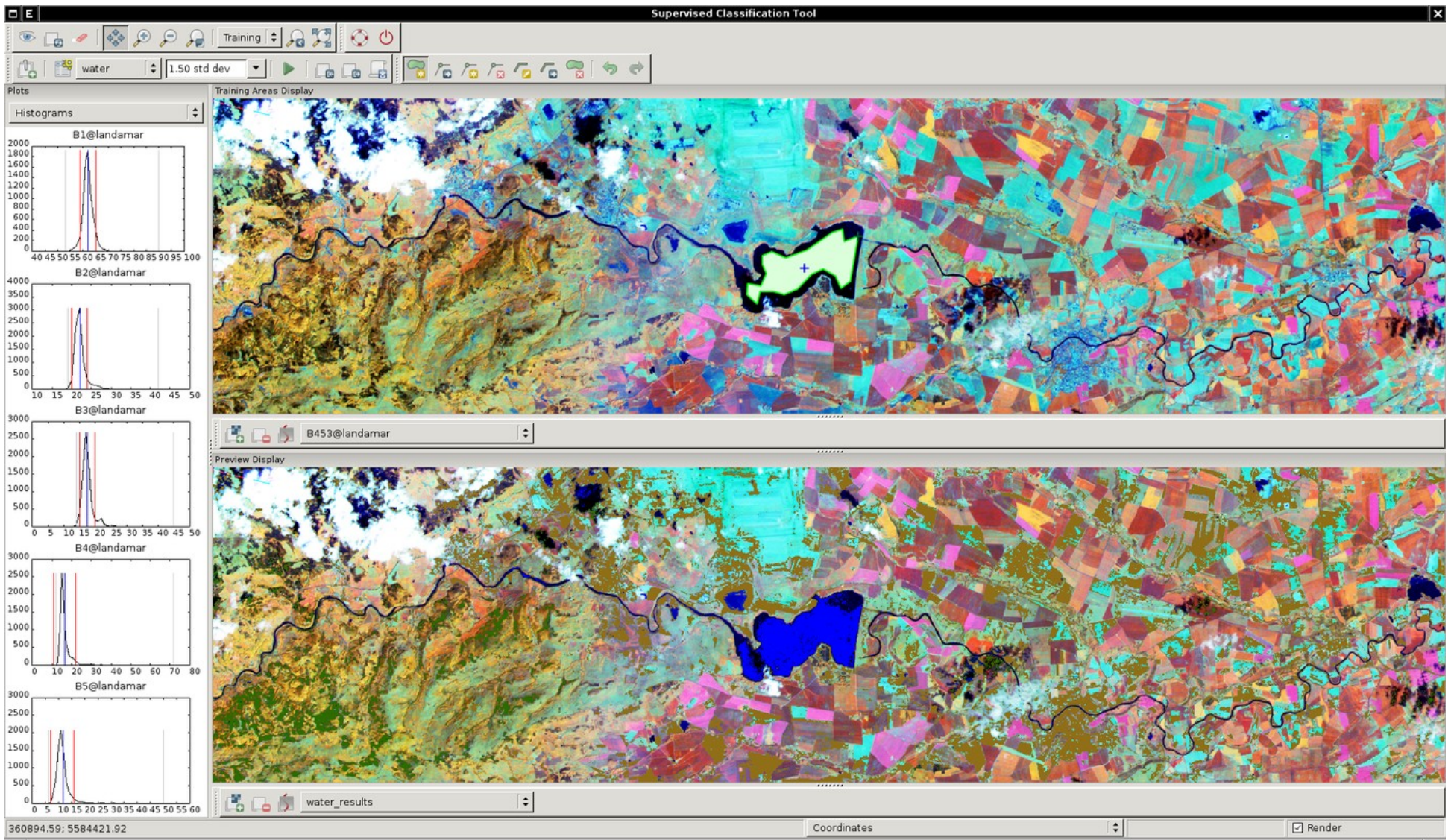
```
i.segment group=ortho2010_t792_subset_20cm \  
  output=ortho2010_t792_subset_20cm_segment_50 \  
  seeds=ortho2010_t792_subset_20cm_segment_25 \  
  goodness=ortho2010_t792_subset_20cm_seg_50_fit \  
  threshold=0.50
```



Supervised image classification



<http://geo.fsv.cvut.cz/~landa/publications/2012/ogrs2012/poster/figures/>

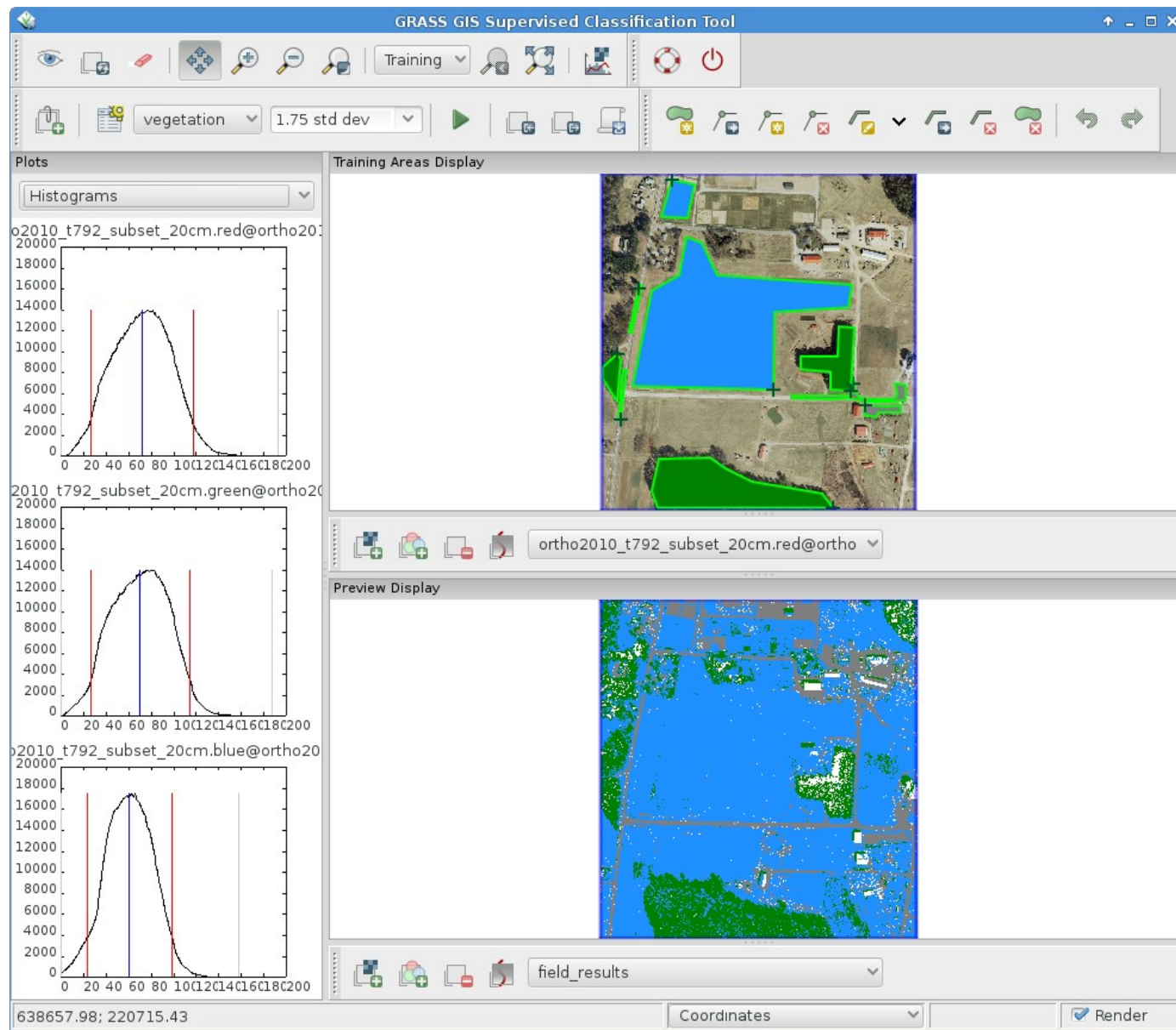


Tool for supervised classification of imagery data.

Generates spectral signatures for an image by allowing the user to outline regions of interest.



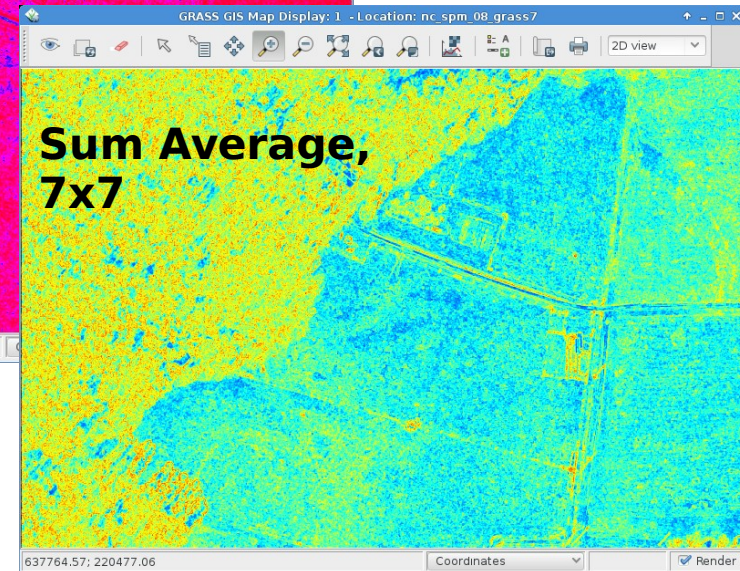
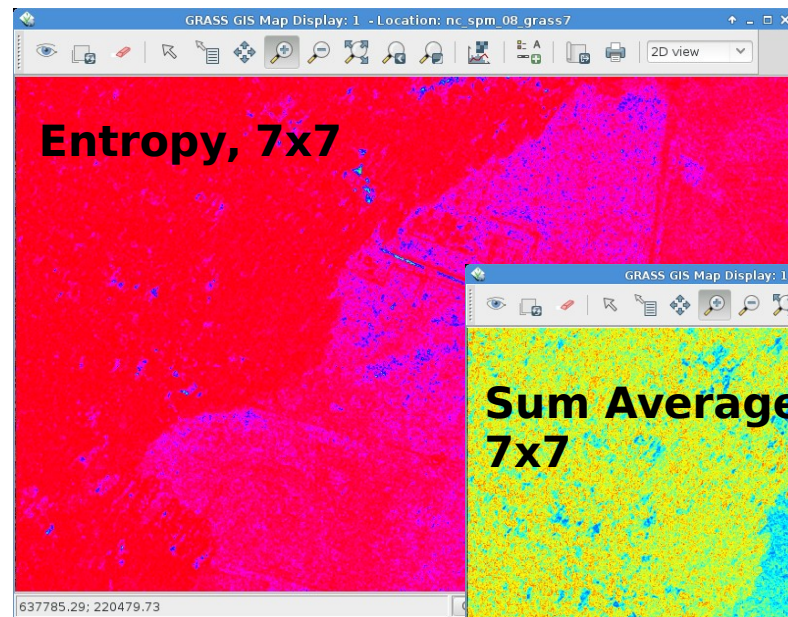
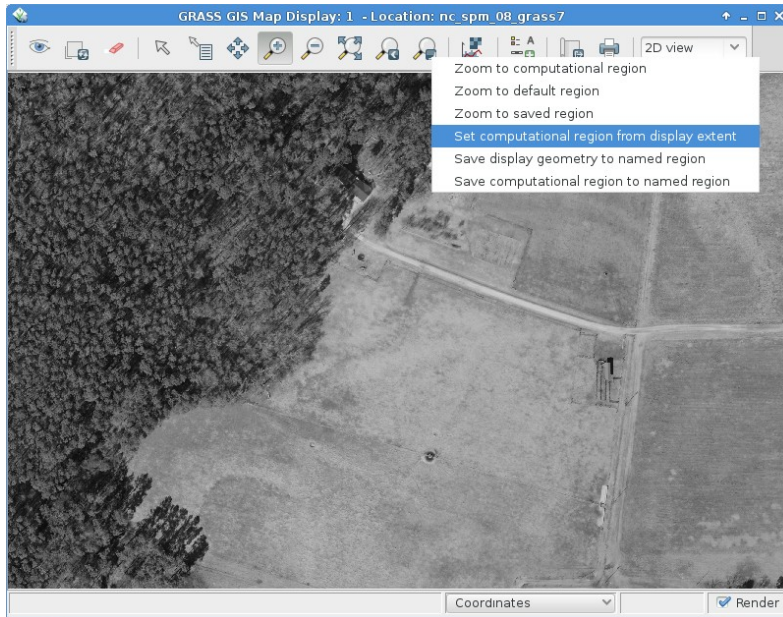
Supervised image classification



Tool for supervised classification of imagery data.
Generates spectral signatures for an image by allowing the user to outline regions of interest. Maxlik Classification and storage as separate raster map per class.



Texture analysis



r.texture

Calculation of the **sum average** (SA) and of the **entropy** (entr) measure, using 7x7 moving window size

Texture maps may be used as synthetic channels for image classification

Space-Time functionality in GRASS 7



Temporal data processing in GRASS GIS

The temporal GIS framework in GRASS introduces three new datatypes that are designed to handle time series data:

- *Space time raster datasets* (strds) are designed to manage raster map time series. Modules that process strds have the naming prefix *t.rast*.
- *Space time 3D raster datasets* (str3ds) are designed to manage 3D raster map time series. Modules that process str3ds have the naming prefix *t.rast3d*.
- *Space time vector datasets* (stvds) are designed to manage vector map time series. Modules that process stvds have the naming prefix *t.vect*.

Temporal data management in general

List of general management modules:

- [t.connect](#)
- [t.create](#)
- [t.remove](#)
- [t.register](#)
- [t.unregister](#)
- [t.info](#)
- [t.list](#)
- [t.rast3d.list](#)
- [t.vect.list](#)
- [t.vect.db.select](#)
- [t.sample](#)
- [t.support](#)
- [t.topology](#)

Export/import conversion

- [t.rast.export](#)
- [t.rast.import](#)
- [t.rast.out.vtk](#)
- [t.rast.to.rast3](#)
- [r3.out.netcdf](#)
- [t.vect.export](#)

Querying and map calculation

- [t.rast.list](#)
- [t.rast.extract](#)
- [t.rast.gapfill](#)
- [t.rast.mapcalc](#)
- [t.rast3d.extract](#)
- [t.rast3d.mapcalc](#)
- [t.rast3d.univar](#)
- [t.vect.extract](#)
- [t.vect.import](#)
- [t.vect.observe.strds](#)
- [t.vect.univar](#)
- [t.vect.what.strds](#)

Aggregation

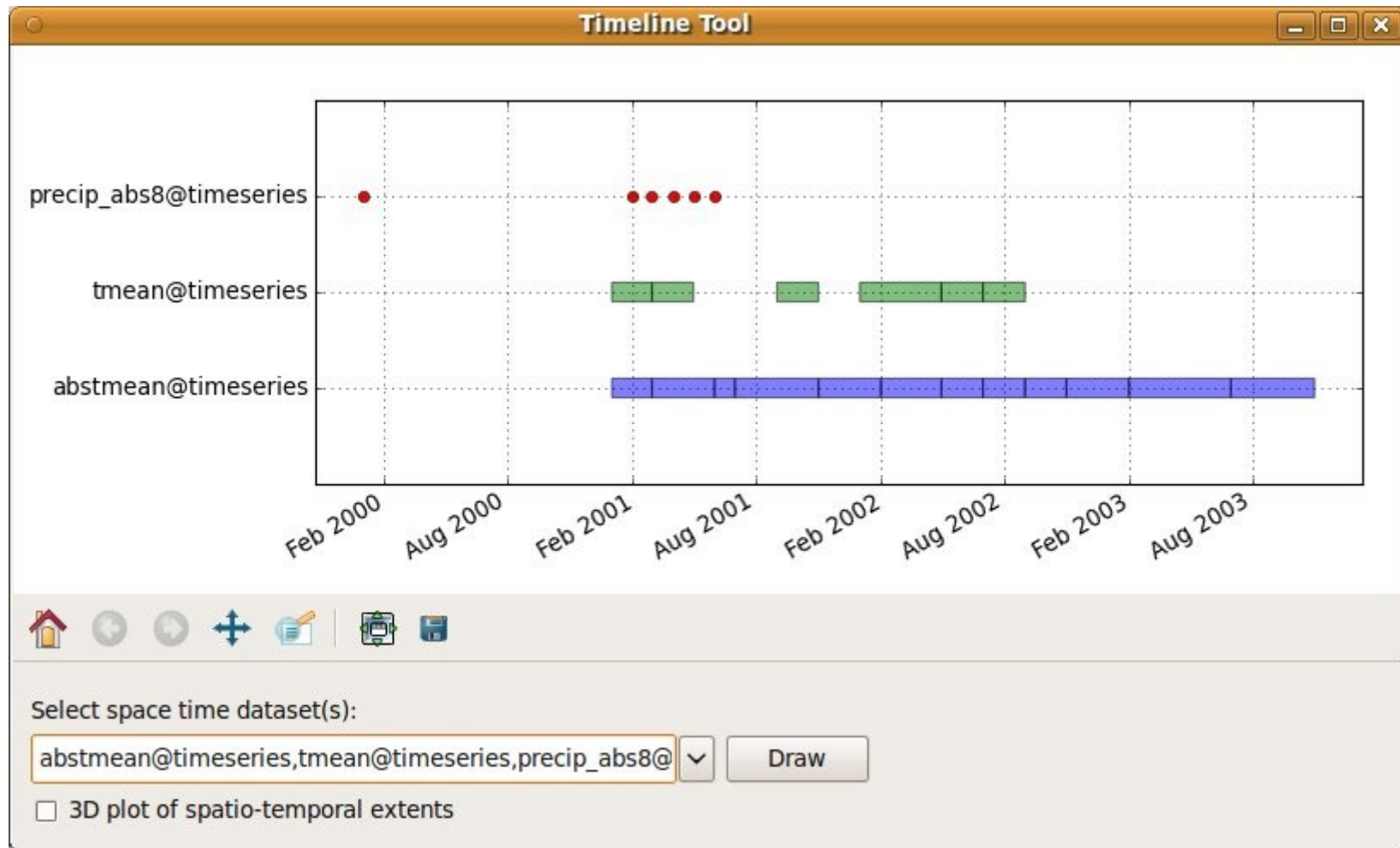
- [t.rast.aggregate.ds](#)
- [t.rast.aggregate](#)
- [t.rast.series](#)

Statistics and gap filling

- [t.rast.gapfill](#)
- [t.rast.univar](#)

Space time datasets are stored in a temporal database. SQLite3 or PostgreSQL are supported as SQL database back end. Connection settings are performed with [t.connect](#). As default a sqlite3 database will be created in the PERMANENT mapset that stores all space time datasets and registered time series maps from all mapsets in the location.

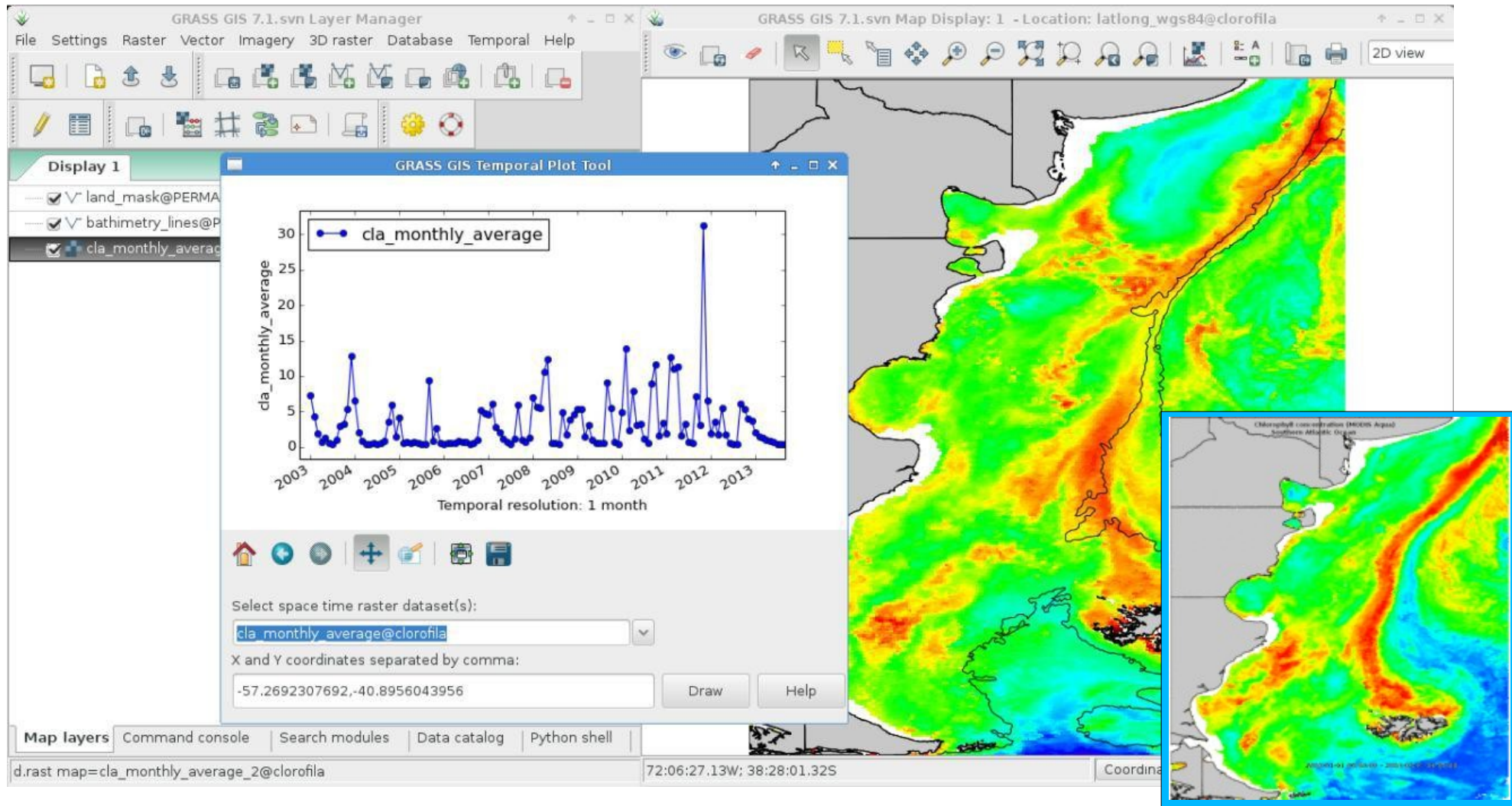
GRASS GIS 7: Space-time functionality



Screenshot: S Gebbert/A. Petrasova

t.register: Registers raster, vector and raster3d maps in a space time dataset

GRASS GIS 7: Space-time functionality



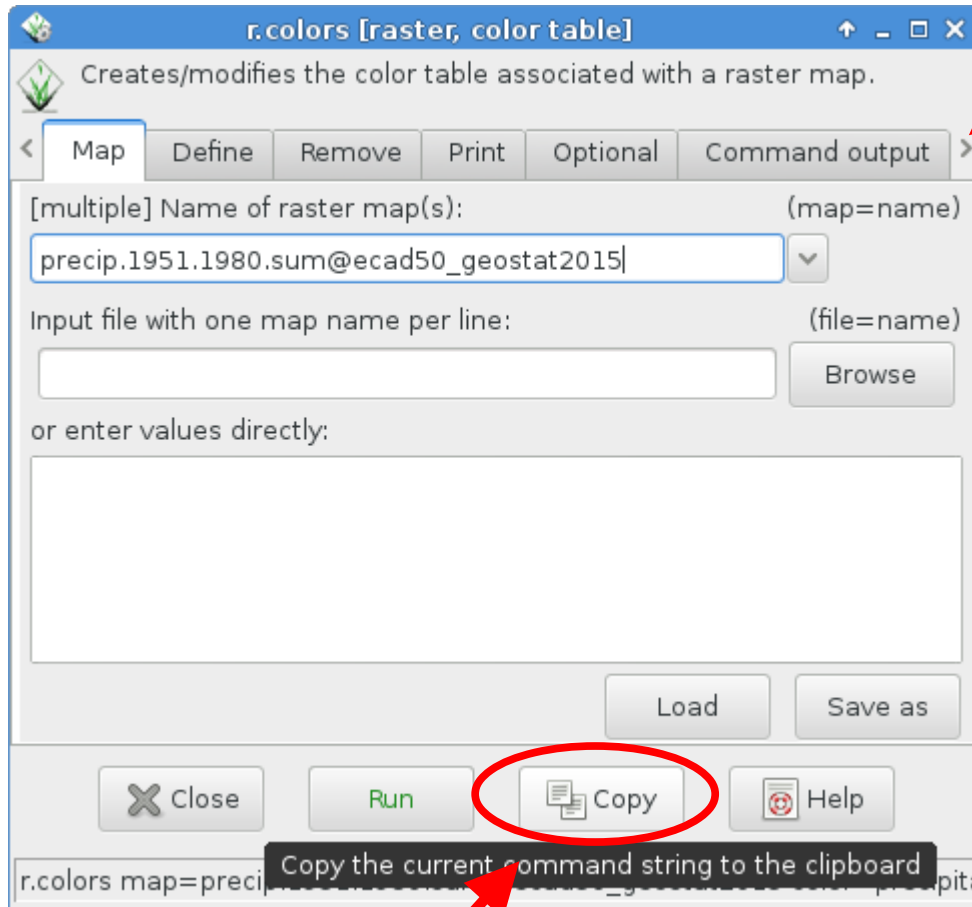
g.gui.tplot: plots the values of one or more temporal raster datasets for a queried point defined by a coordinate pair

(in PDF, click for animation)

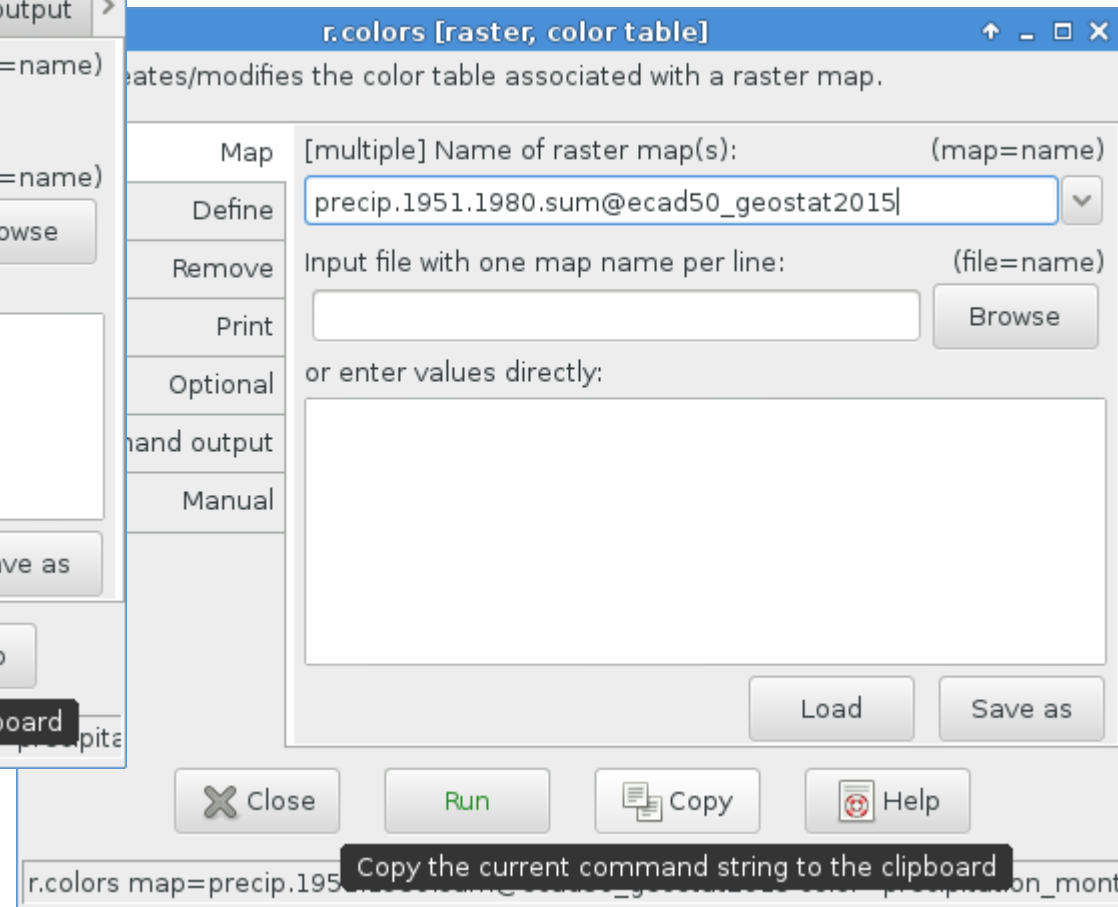
Graphical user interface versus Command line



STYLE: Menu: Settings → GUI Settings → Appearance → Module dialog style: Basic top/left



Cycle through the various tabs



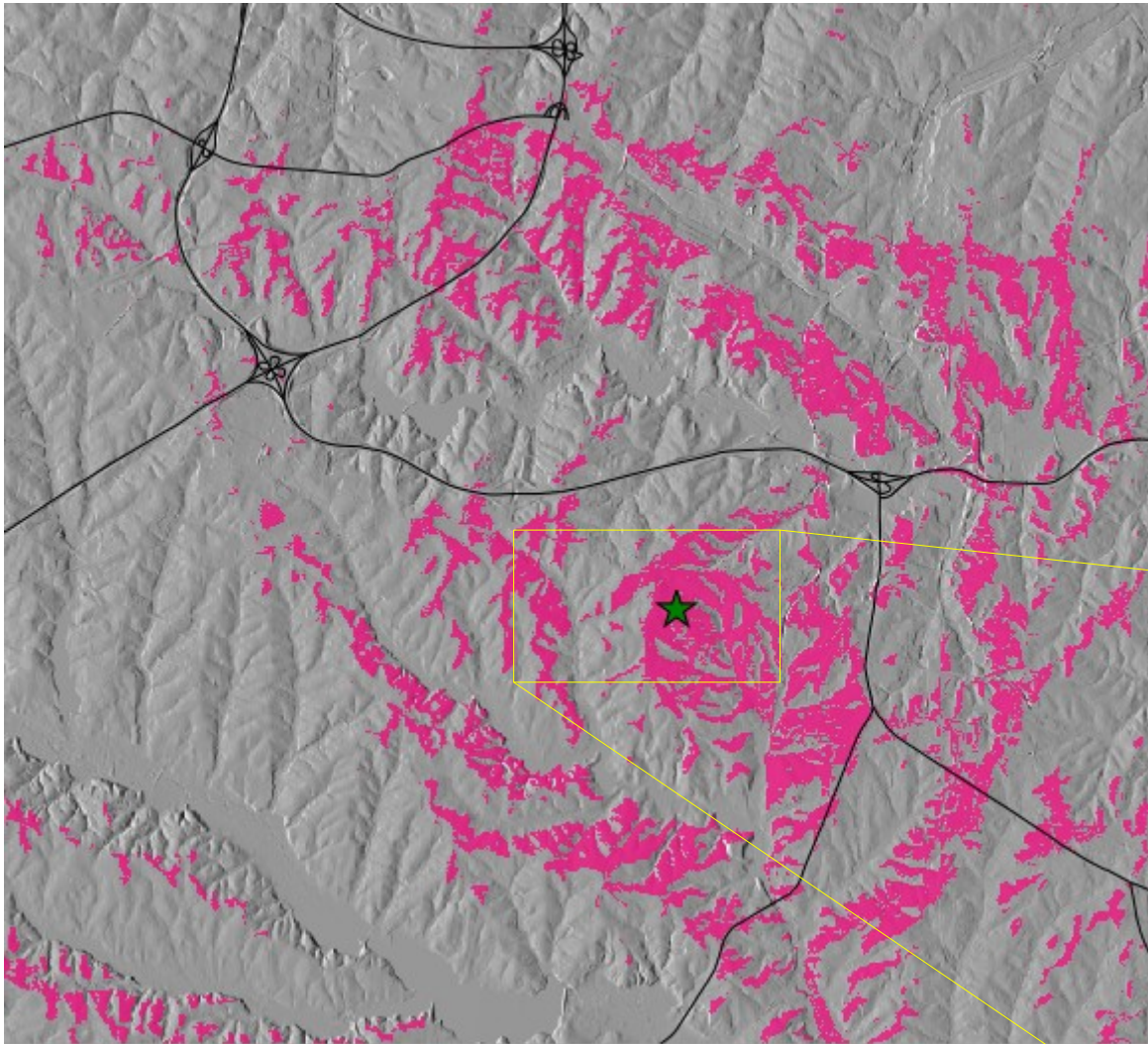
You may copy the command to your documentation

The graphical user interface effectively generates the respective command for the command line (and also writes to the shell “history”)



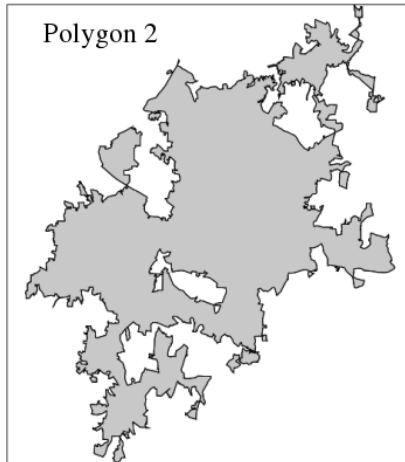
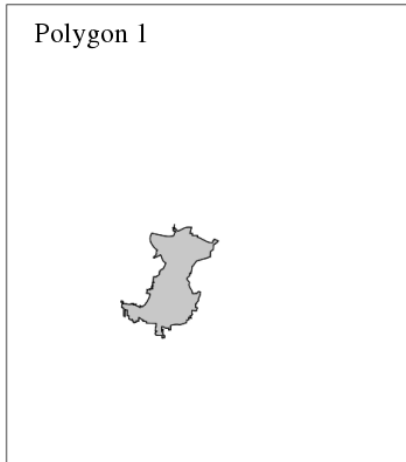
Viewshed (line of sight)

`r.viewshed` in action: what is visible from a certain viewpoint?





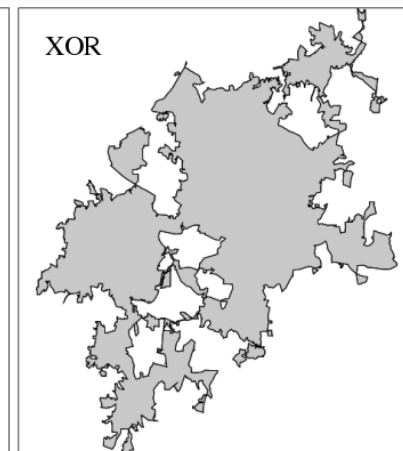
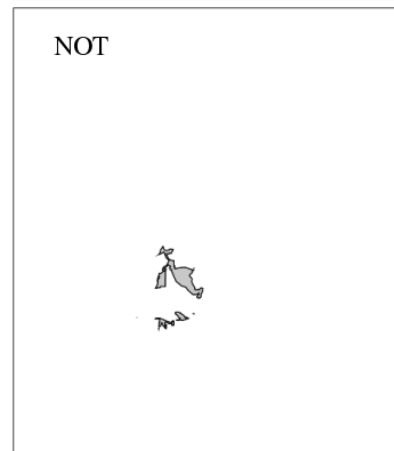
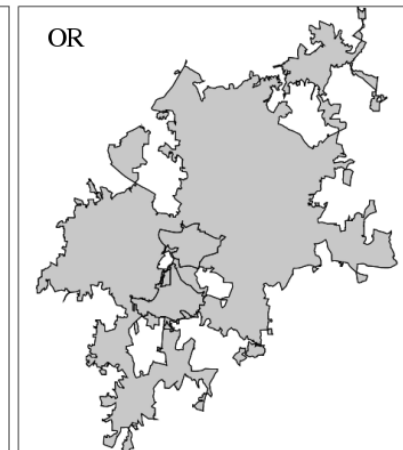
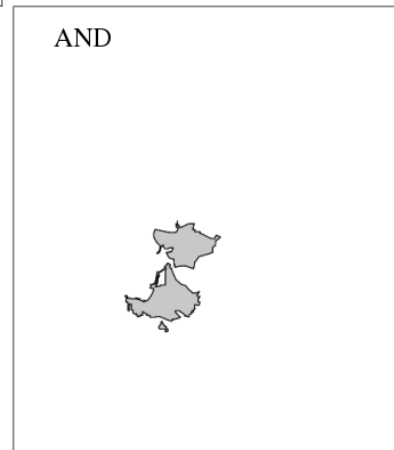
Vector feature overlay operations



Boolean operators

GRASS GIS module:

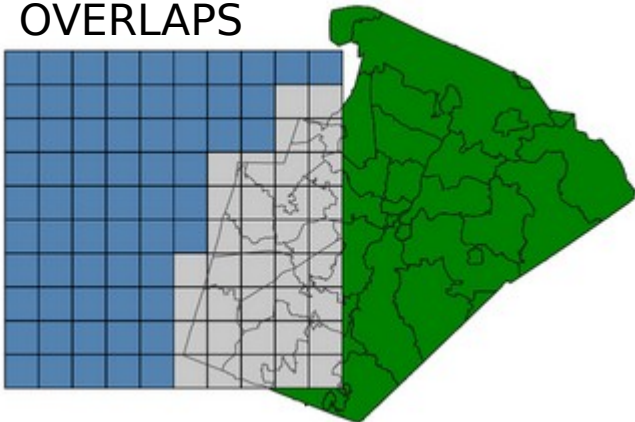
`v.overlay`



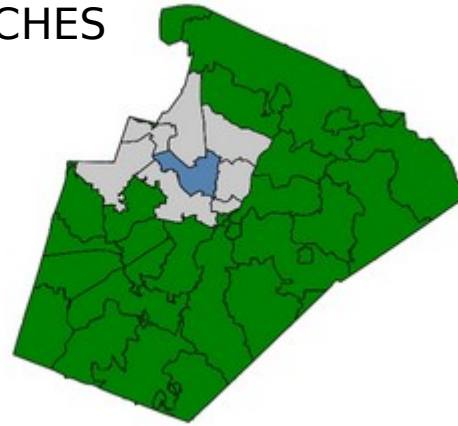
Vector feature select operations: v.select (GEOS)



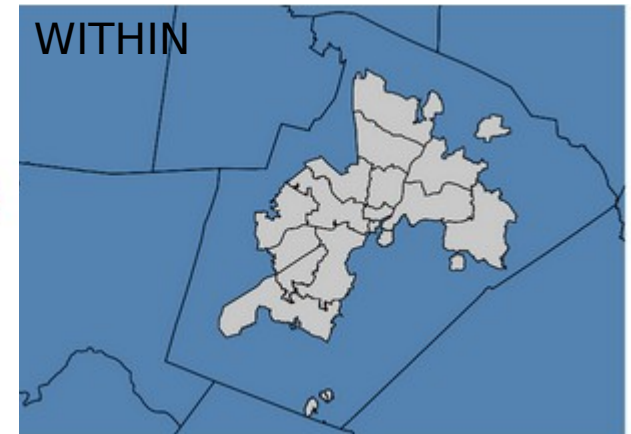
OVERLAPS



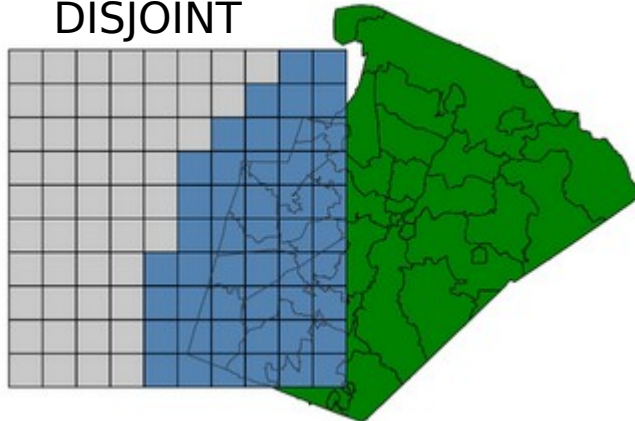
TOUCHES



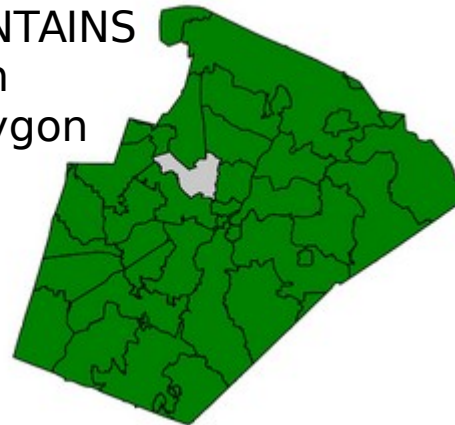
WITHIN



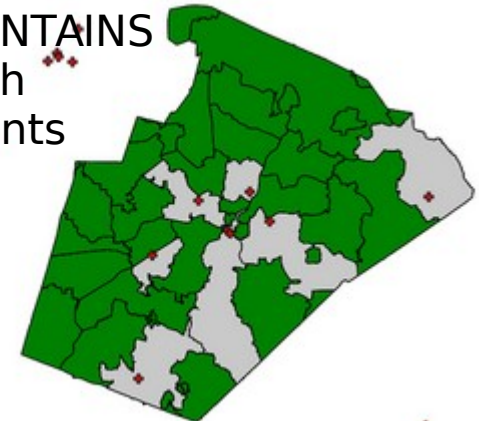
DISJOINT



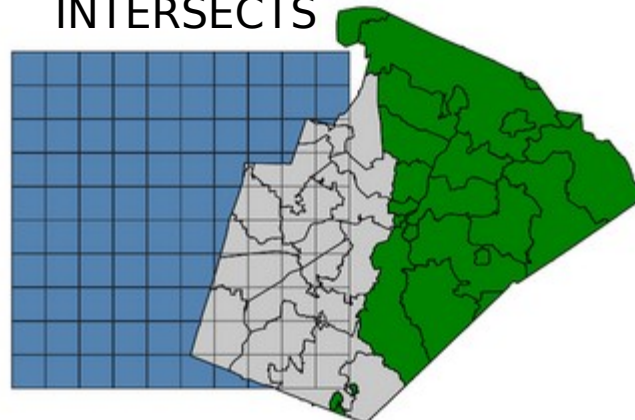
CONTAINS
with
polygon



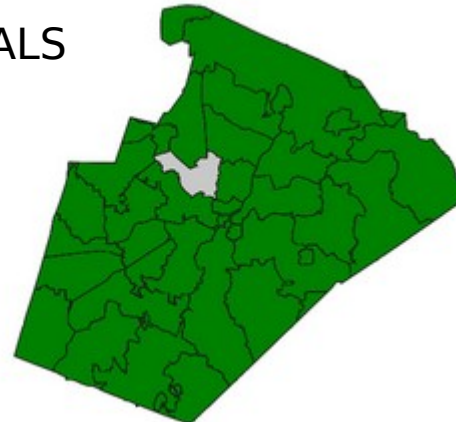
CONTAINS
with
points



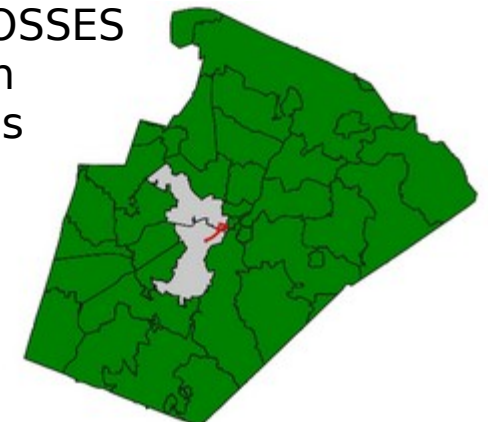
INTERSECTS



EQUALS



CROSSES
with
lines



Overview: Vector network analysis in GRASS GIS



GRASS GIS 7.0.0svn Layer Manager

File Settings Raster Vector Imagery Volumes Database Temporal Help

Develop vector map
Topology maintenance
Manage colors
Query vector map
Feature selection
Map type conversions

Display 1

Network analysis

Vector network analysis tool
Network preparation [v.net]
Allocate subnets [v.net.alloc]
Split net [v.net.iso]
Shortest path [v.net.path]
Shortest path for sets of features [v.net.distance]
Shortest path using timetables [v.net.timetable]
Shortest path for all pairs [v.net.allpairs]
Visibility network [v.net.visibility]
Bridges and articulation points [v.net.bridge]
Maximum flow [v.net.flow]
Vertex connectivity [v.net.connectivity]
Components [v.net.components]
Centrality [v.net.centrality]
Steiner tree [v.net.steiner]
Minimum spanning tree [v.net.spanningtree]
Traveling salesman analysis [v.net.salesman]

Map layers Command console Search modules Python shell



Vector network analysis in GRASS GIS

Procedure: Display the vector network, activate snapping to nodes (takes a moment) and define two points on the network

The image shows a screenshot of the GRASS GIS 7.0.0svn interface. The main window is titled "GRASS GIS 7.0.0svn Map Display: 1 - Location: nc_spm". The "GRASS GIS Vector Network Analysis Tool" is open, showing the "Shortest path (v.net.path)" tool. The "Points for analysis:" section is active, with a tooltip that says "Activate snapping to nodes" and a red number "4" pointing to the snapping icon. The tool is set to "topology" and has two points defined: "1 Start point new point" and "2 End point new point". A red number "3" is placed near the "nc_streets_wake_major" layer in the Layer Manager. A second, larger inset shows the same tool with a tooltip that says "Insert points from Map Display" and a red number "5" pointing to the "Insert points from Map Display" icon. The map display shows a dense network of black lines representing streets. The status bar at the bottom shows the coordinate "213869.34" and "Coordir".

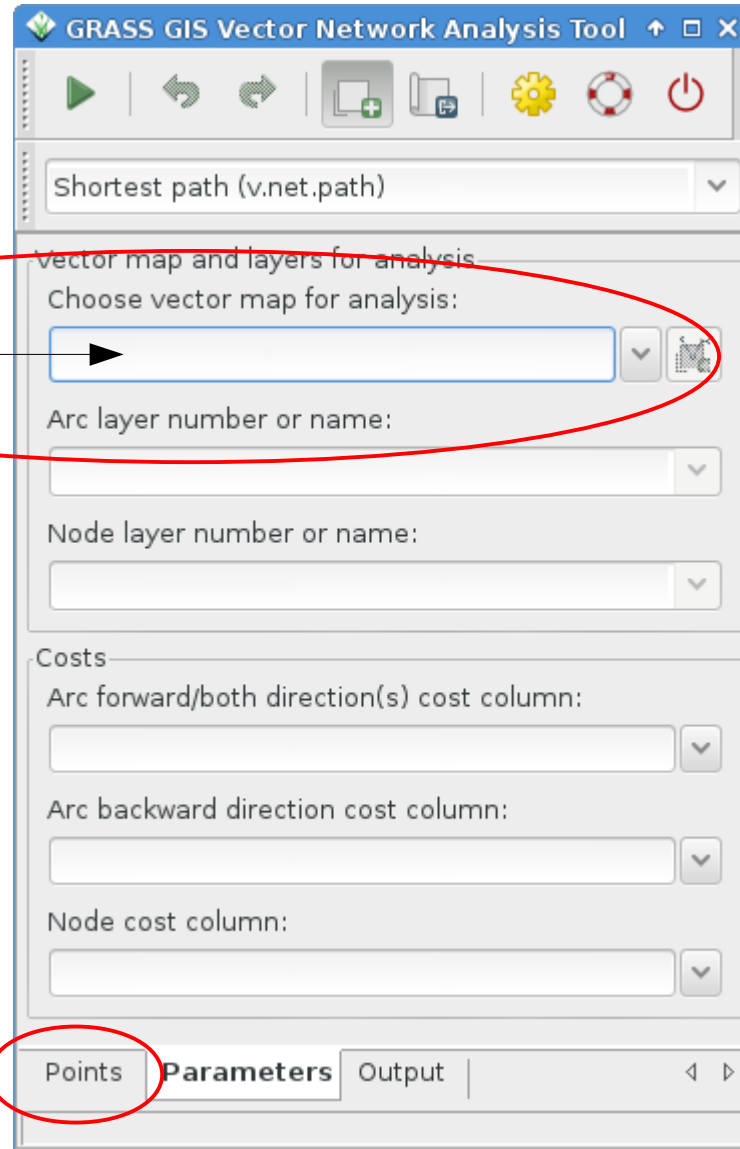
Vector network analysis in GRASS GIS



Vector Network Analysis Tool

1

roads



2

Vector network analysis in GRASS GIS



Vector Network Analysis Tool: shortest path

GRASS GIS Vector Network Analysis Tool

GRASS GIS 7.0.0svn Map Display: 1 - Location: spearfish60_grass7@user1

Execute analysis

Shortest path (v.net.path)

Points for analysis:

use	type	topology
<input checked="" type="checkbox"/>	1 Start point	new point
<input checked="" type="checkbox"/>	2 End point	new point

Analysis settings:

Maximum distance of point to the network: 100000

Points Parameters Output Input tables

589726.65; 4922563.33 Coordinates Render



Vector network analysis in GRASS GIS

Vector Network Analysis Tool: subnet allocation

GRASS GIS Vector Network Analysis Tool

Subnets for nearest centers (v.net.alloc)

Points for analysis:

use topology

- 1 new point
- 2 new point

Analysis settings:

Maximum distance of point to the network: 100000

Points Parameters Output Input tables

GRASS GIS 7.0.0svn Map Display: 1 - Location: spearfish60_grass7@user1

593117.07; 4919367.45

Coordinates

Render



Vector network analysis in GRASS GIS

Vector Network Analysis Tool: cost isolines: 1km, 2km, 3km, 10km

GRASS GIS Vector Network Analysis Tool

Cost isolines (v.net.iso)

Points for analysis:

use topology

- 1 new point
- 2 new point
- 3 new point

Analysis settings:

Maximum distance of point to the network: 100000

Iso lines: 1000,2000,3000,10000

Points Parameters Output Input tables

589476.54; 4917366.55 Coordinates Render



Vector network analysis in GRASS GIS

Vector Network Analysis Tool: travelling salesman, 4 points to visit

The screenshot displays the GRASS GIS Vector Network Analysis Tool interface. The main window shows a map display with a road network and a red path representing the solution to a traveling salesman problem. The path starts at a green cross labeled '1' and visits four points marked with red crosses labeled '1', '2', '3', and '4' in sequence. The interface includes a toolbar at the top, a dropdown menu for the tool name 'Traveling salesman (v.net.salesman)', and a panel for 'Points for analysis' with four points listed and checked. The 'Analysis settings' panel shows a 'Maximum distance of point to the network' set to 100000. The bottom status bar displays coordinates '596507.48; 4920590.22' and a 'Render' checkbox.

GRASS GIS Vector Network Analysis Tool

GRASS GIS 7.0.0svn Map Display: 1 - Location: spearfish60_grass7@user1

Traveling salesman (v.net.salesman)

Points for analysis:

use topology

- 1 new point
- 2 new point
- 3 new point
- 4 new point

Analysis settings:

Maximum distance of point to the network: 100000

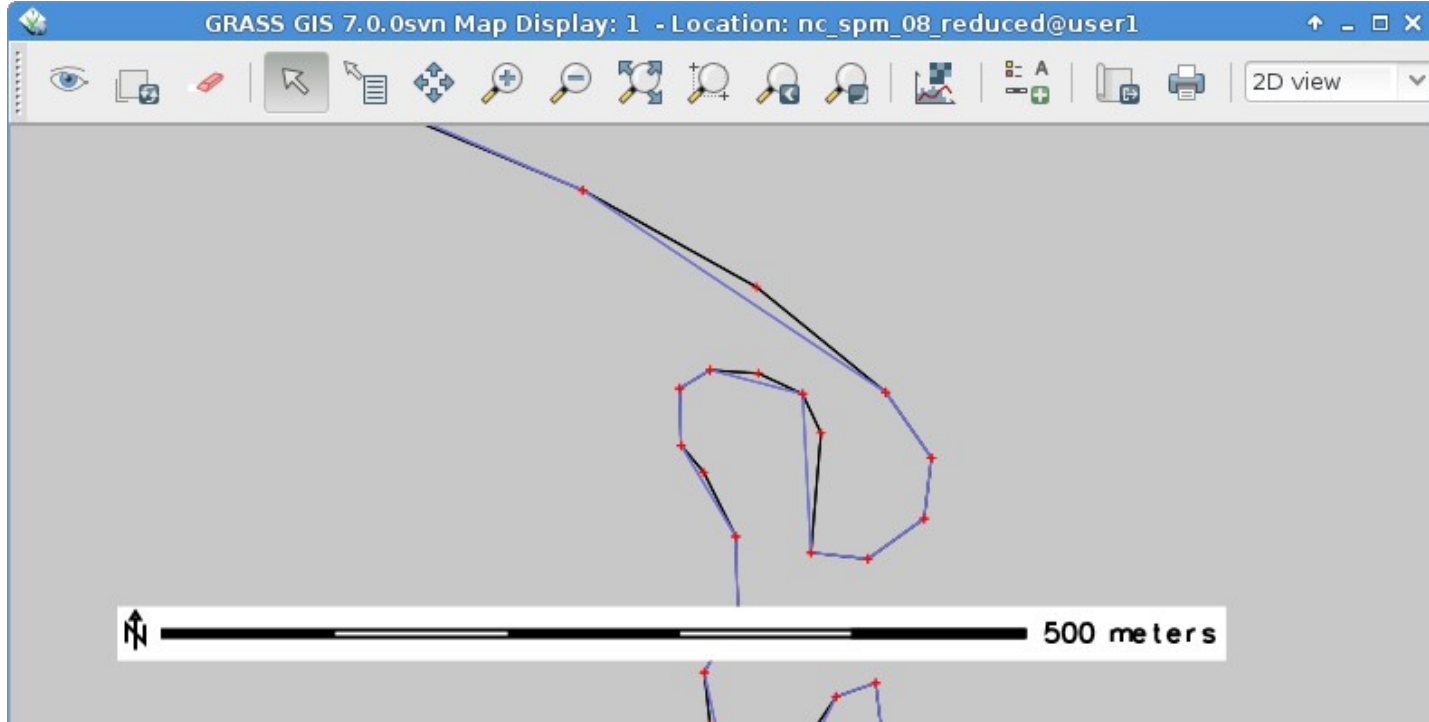
Points Parameters Output Input tables

596507.48; 4920590.22 Coordinates Render



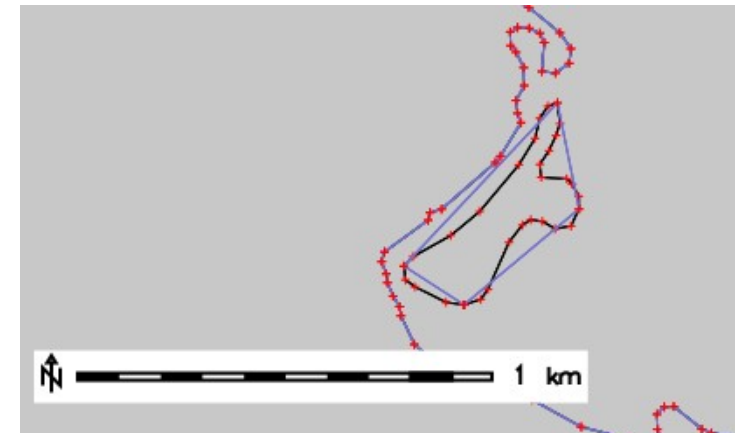
Vector simplification with v.generalize

Reducing the number of vertices with Douglas-Peucker



10m threshold

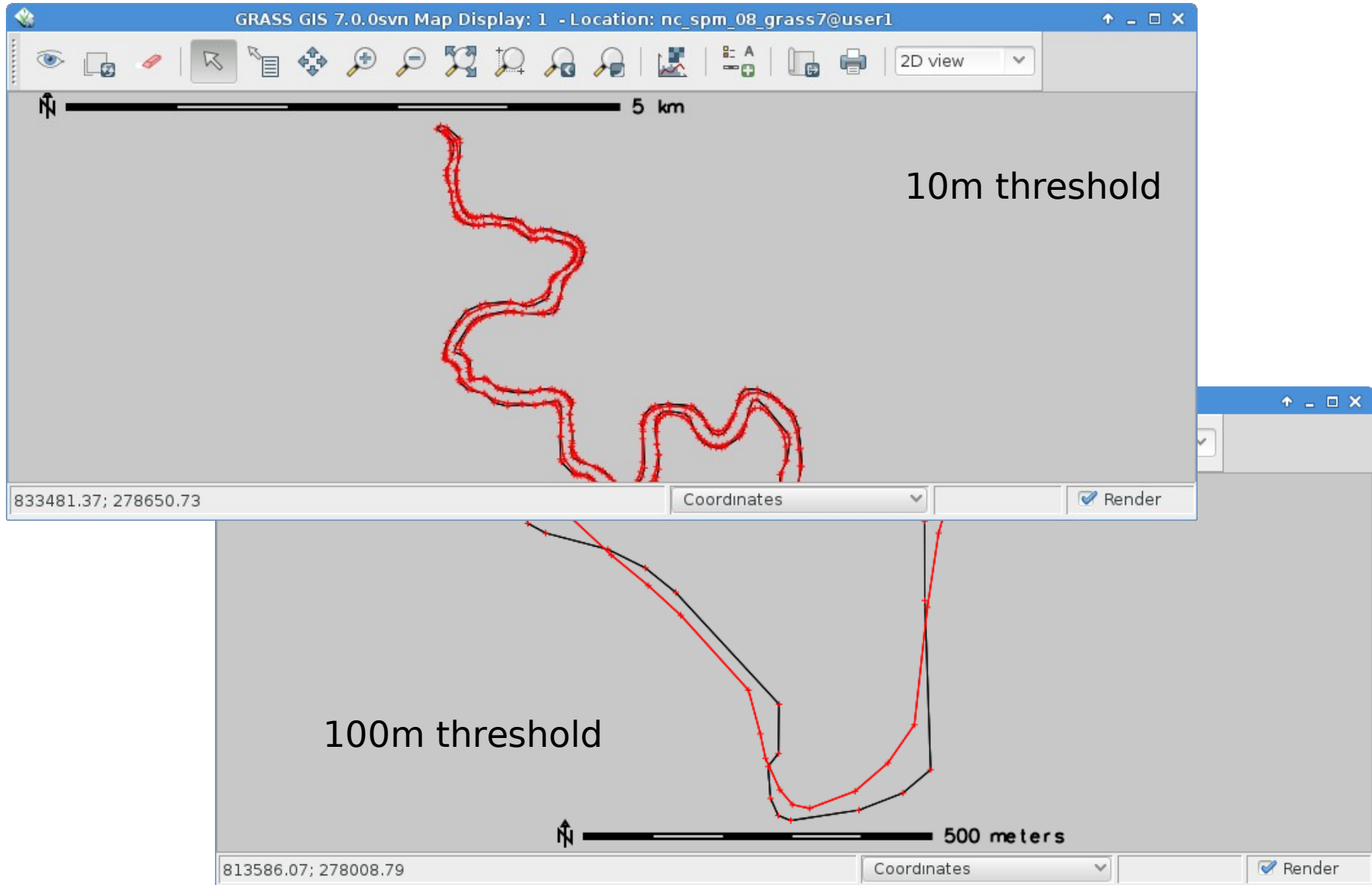
100m threshold





Vector smoothing with v.generalize

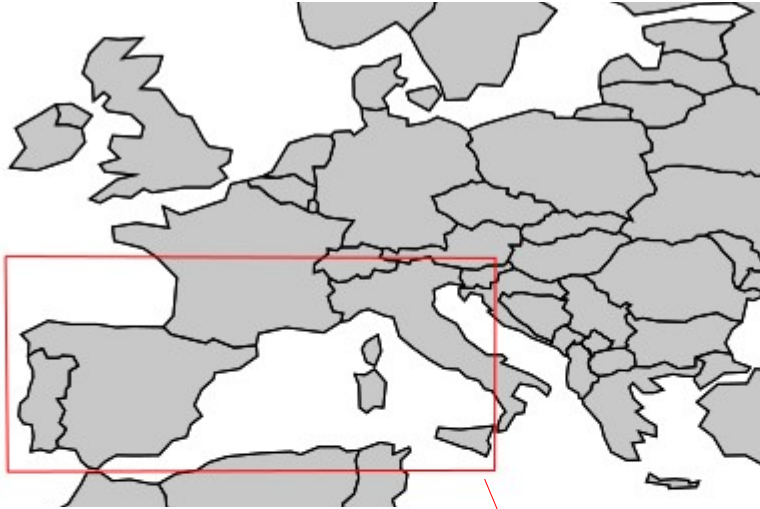
Increasing the number of vertices with sliding average





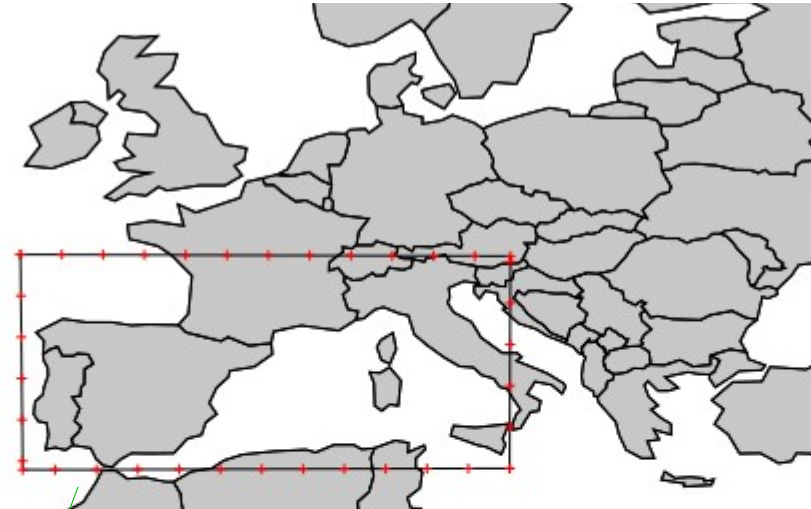
Vector reprojection: do it right!

A rectangular in LatLong being reprojected...



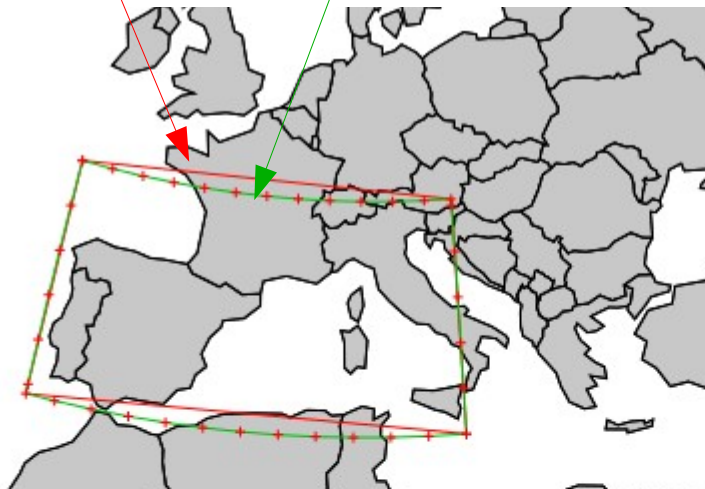
4 corner rectangle, LatLong
(no further vertices there; an
issue in many GIS)

Ouch...



**Automated vertex densification
in GRASS GIS 7 (v.proj)**

OK



...to EU LAEA

New Python API integration



http://grass.osgeo.org/wiki/GRASS_and_Python



Page [Discussion](#)

Read [Edit](#) [View history](#)

GRASS and Python

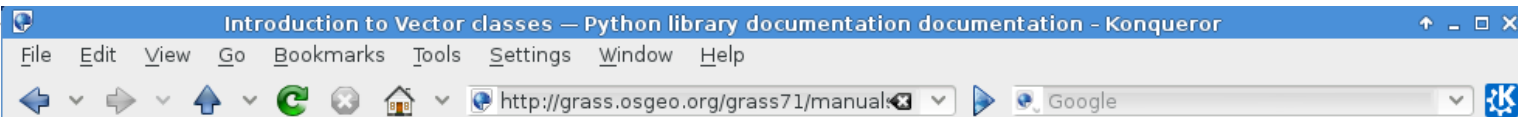
Navigation

[Main Page](#)
[Community](#)
[Development](#)
[Documents](#)
[GRASS Help](#)
[Recent changes](#)
[Help](#)

Toolbox

[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)

- 1 [Python SIGs](#)
- 2 [Writing Python scripts](#)
 - 2.1 [Python script ed](#)
 - 2.2 [Using the GRA](#)
 - 2.3 [Creating Python](#)
 - 2.3.1 [MS-Wind](#)
 - 2.3.2 [Linux](#)
 - 2.4 [Running extern](#)
 - 2.5 [Testing and inst](#)
 - 2.5.1 [Debuggin](#)
 - 2.5.2 [Installatic](#)
- 3 [Python extensions in C](#)
 - 3.1 [Python Scripting](#)
 - 3.2 [Python Ctypes I](#)
 - 3.3 [wxPython GUI c](#)
 - 3.4 [Python-GRASS](#)
 - 3.5 [Using GRASS g](#)
- 4 [FAQ](#)
- 5 [Links](#)
 - 5.1 [General guides](#)
 - 5.2 [Programming](#)
 - 5.3 [Presentations](#)



[Python library documentation documentation](#) » [PyGRASS documentation](#) »

[previous](#) | [next](#) | [modules](#) | [index](#)

Introduction to Vector classes

Details about the architecture can be found in the [GRASS GIS 7 Programmer's Manual: GRASS Vector Library](#)

Instantiation and basic interaction.

```
>>> from pygrass.vector import VectTopo
>>> municip = VectTopo('boundary_municip_sqlite')
>>> municip.is_open()
False
>>> municip.mapset
''
>>> municip.exist() # check if exist, and if True set mapset
True
>>> municip.mapset
'user1'
```

Open the map with topology:

```
>>> municip.open()
```

get the number of primitive:

Previous topic

[Introduction to Raster classes](#)

Next topic

[Interface to GRASS GIS modules](#)

Quick search

Enter search terms or a module, class or function name.

Using Python and GRASS GIS 7 with ipython



An interactive (Web based!) shortcourse on writing GRASS scripts in Python

<https://github.com/wenzeslaus/python-grass-addon>

https://github.com/wenzeslaus/python-grass-addon/blob/master/01_scripting_library.ipynb

Introduction to the GRASS GIS 7 Python Scripting Library

The [GRASS GIS 7](#) Python Scripting Library provides functions to call GRASS modules within scripts as subprocesses. The most often used functions include:

- **run_command**: most often used with modules which output raster/vector data where text output is not expected
- **read_command**: used when we are interested in text output
- **parse_command**: used with modules producing text output as key=value pair
- **write_command**: for modules expecting text input from either standard input or file

Besides, this library provides several wrapper functions for often called modules.

Calling GRASS GIS modules

We start by importing GRASS GIS Python Scripting Library:

```
In [ ]: import grass.script as gscript
```

Before running any GRASS raster modules, you need to set the computational region using [g.region](#). In this example, we set the computational extent and resolution to the raster layer `elevation`.

```
In [ ]: gscript.run_command('g.region', raster='elevation')
```

The `run_command()` function is the most commonly used one. Here, we apply the focal operation *average* ([r.neighbors](#)) to smooth the elevation raster layer. Note that the syntax is similar to bash syntax, just the flags are specified in a parameter.

GRASS GIS goes supercomputer: HPC



- Since **2005** (10 years) GRASS GIS is running **natively** on 64bit CPUs
- GRASS GIS 7 offers Large File Support also on 32bit Windows
- Installed on Grids and TOP500 supercomputers (AKKA Umeå, ENEA Frascati, Aurel Bratislava, ...)
- Runs on Linux, AIX, Solaris, freeBSD, netBSD, (MS-Windows)...
- Various ways of parallelization



Hints: http://grasswiki.osgeo.org/wiki/Compile_and_Install

GRASS GIS 7 and R statistics



Using R (Rstudio) within a GRASS GIS session

https://grasswiki.osgeo.org/wiki/R_statistics/rgrass7

```
neteler@oboe:~  
  
Welcome to GRASS GIS 7.0.2svn (r65960)  
GRASS GIS homepage: http://grass.osgeo.org/  
This version running through: Bash Shell  
Help is available with the command: g.manual  
See the licence terms with: g.version  
Start the GUI with: g.gui  
When ready to quit enter: exit  
  
GRASS 7.0.2svn (nc_spm_08_grass7):~ > rstudio &  
[1] 26735  
GRASS 7.0.2svn (nc_spm_08_grass7):~ >   
  
RStudio  
File Edit Code View Plots Session Build Debug Tools Help  
Go to file/function Clearing plots...  
Console ~/   
R version 3.2.1 (2015-06-18) -- "World-Famous Astronaut"  
Copyright (C) 2015 The R Foundation for Statistical Computing  
Platform: x86_64-redhat-linux-gnu (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> library(rgrass7)  
Loading required package: sp  
Loading required package: XML  
GRASS GIS interface loaded with GRASS version: GRASS 7.0.2svn (2015)  
and location: nc_spm_08_grass7  
> execGRASS("g.region", raster = "elevation", flags = "p")  
projection: 99 (Lambert Conformal Conic)  
zone: 0  
datum: nad83  
ellipsoid: a=6378137 es=0.006694380022900787  
north: 228500
```



Using QGIS with Processing/GRASS

Geoprocessing of the fire point layer: QGIS → Processing → GRASS GIS

The screenshot shows the QGIS 2.2.0-Valmiera interface. The main window displays a map with a hillshade layer and a point layer named 'firms123561400683...'. The Processing Toolbox on the right lists various algorithms, with 'v.voronoi - Create...' selected. The dialog box for 'v.voronoi' is open, showing the following parameters:

- Input points layer: firms123561400683502_mcd14ml [EPSG:4326]
- Output tessellation as a graph (lines), not areas: No
- Do not create attribute table: No
- GRASS region extent(xmin, xmax, ymin, ymax): [Leave blank to use min covering extent]
- Voronoi diagram: [Save to temporary file]
- Open output file after running algorithm:

The 'GRASS region extent' field is circled in red. The status bar at the bottom shows the coordinate as 10.981,46.142 and the scale as 1,206,517.

GRASS Addons: User contributed extensions



The Addons repository is SVN based:

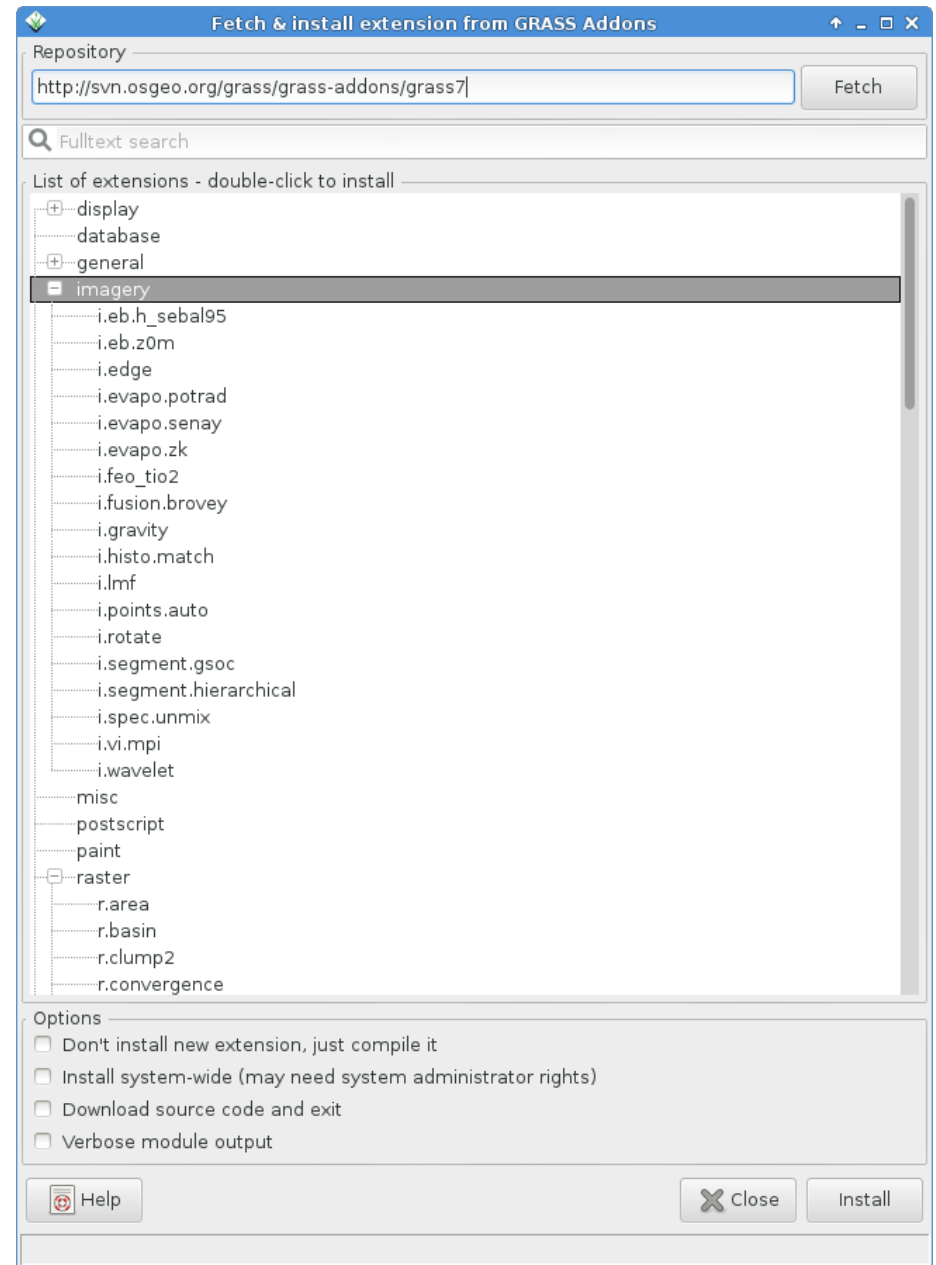
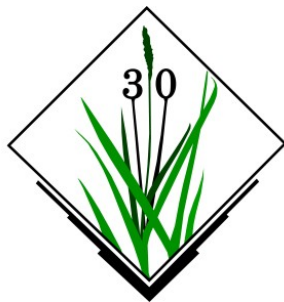
One-click installation with extension manager

Increasing inflow of Python scripts

Users can easily obtain **write** access to develop new functionality

Peer review through SVN commit email list

Also github, gitlab etc. now supported



<https://grass.osgeo.org/grass70/manuals/addons/>

Development meetings: Community sprints



GRASS-GIS Community Sprint 2012, Prague, Czech Republic



Community Sprint in Como 2015



Where's the stuff?



GRASS GIS 7 Software:

Free download for MS Windows, MacOSX, Linux and source code:

<https://grass.osgeo.org/download/>

Addons (user contributed extensions):

<https://grass.osgeo.org/grass70/manuals/addons/>

Free sample data:

Rich data set of North Carolina (NC)

... available as GRASS GIS location and in common GIS formats

<https://grass.osgeo.org/download/sample-data/>

User Help:

Mailing lists (also in different languages):

<https://grass.osgeo.org/support/>

Wiki:

<https://grasswiki.osgeo.org/wiki/>

https://grasswiki.osgeo.org/wiki/R_statistics/rgrass7

Manuals:

<https://grass.osgeo.org/documentation/manuals/>