

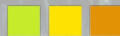
Markus Neteler
Fondazione E. Mach – CRI, Italy
<http://gis.cri.fmach.it>
& GRASS Development Team

News in GRASS GIS 7

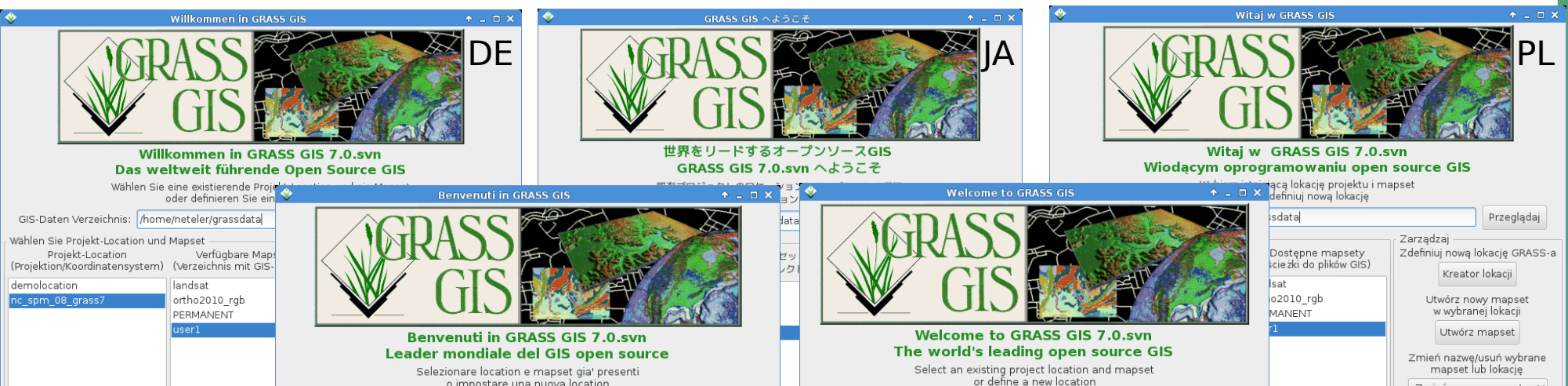
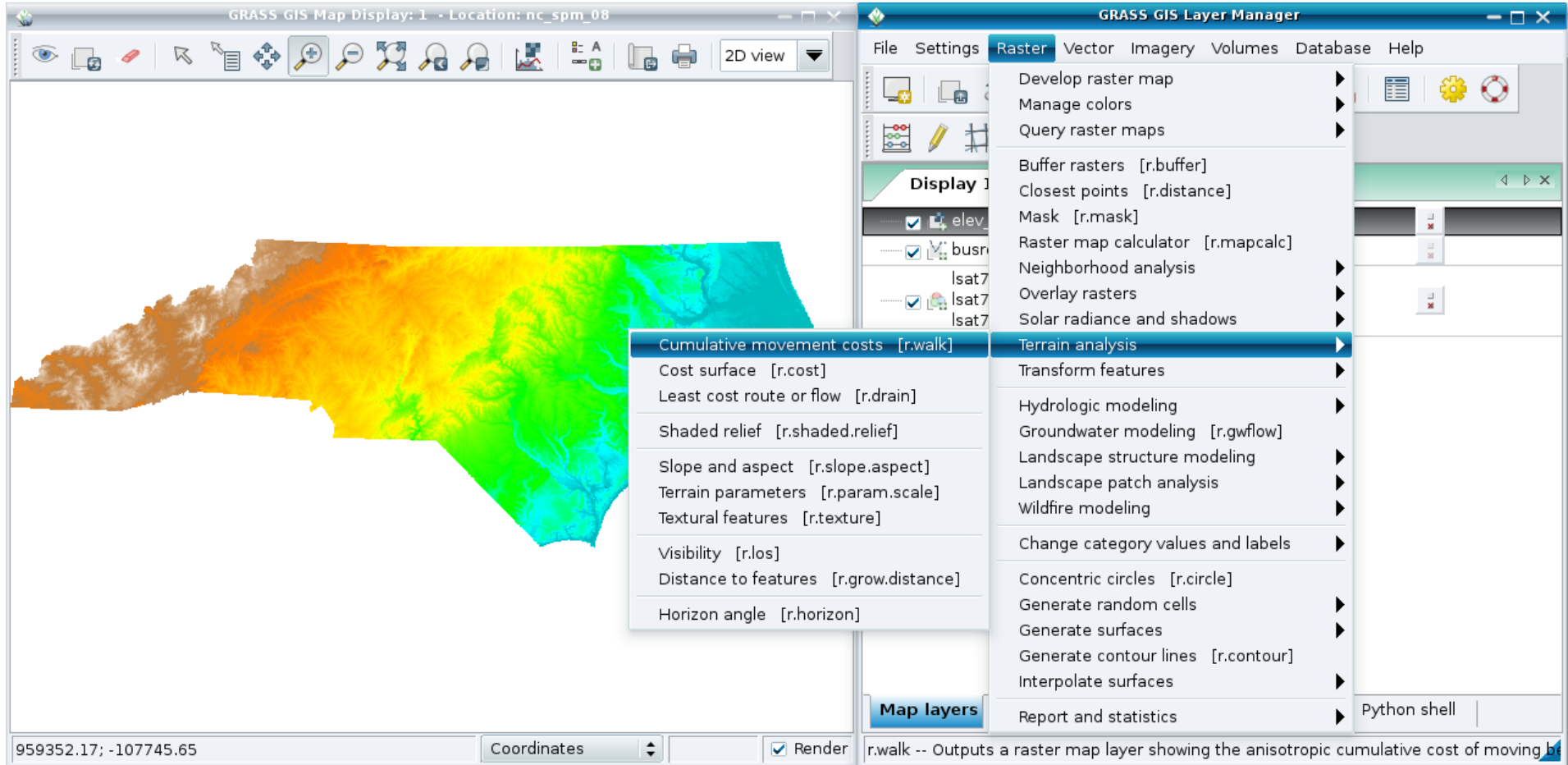
**FOSS4G-CEE 2013
Bucharest, Romania**



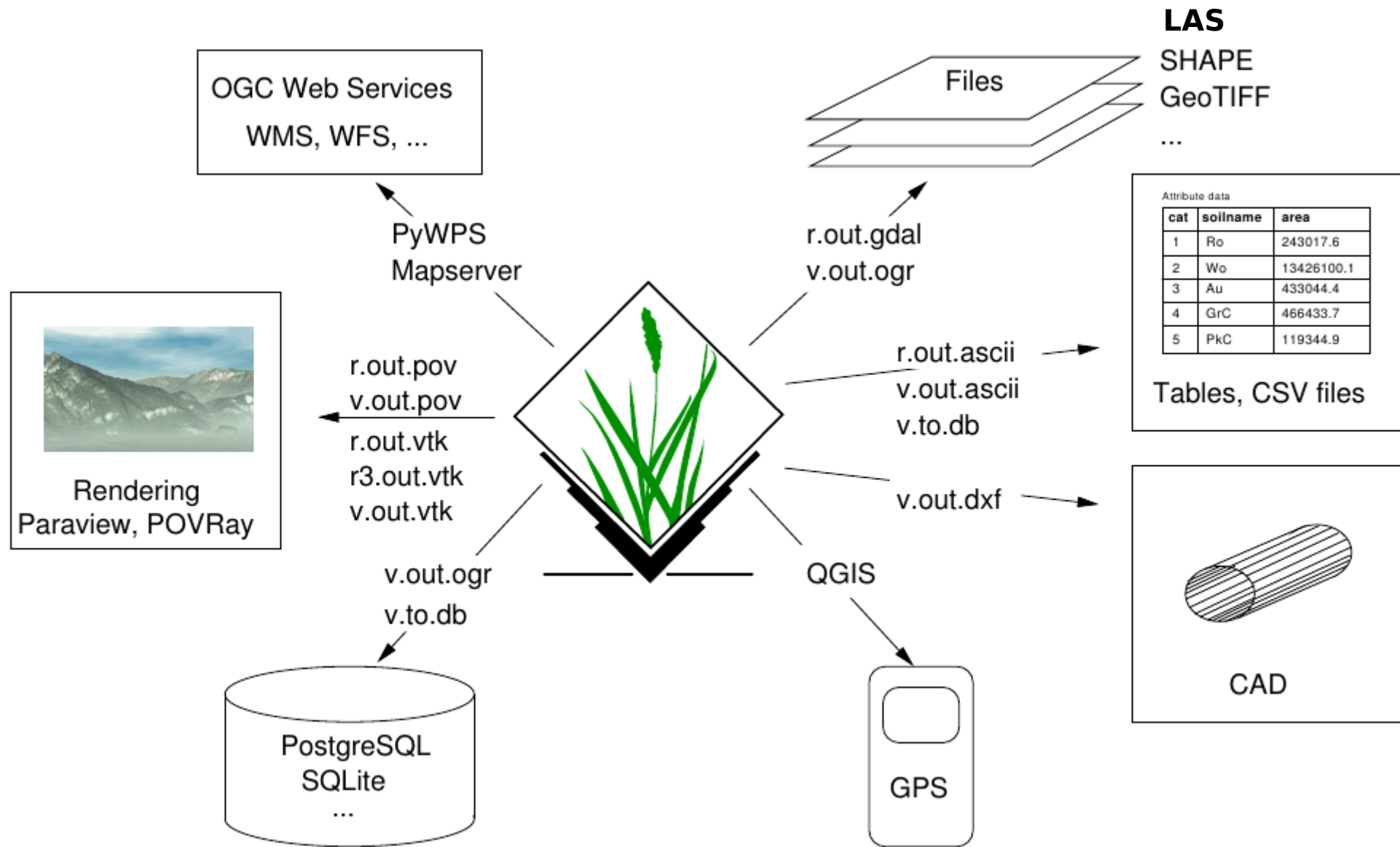
FONDAZIONE
EDMUND
MACH



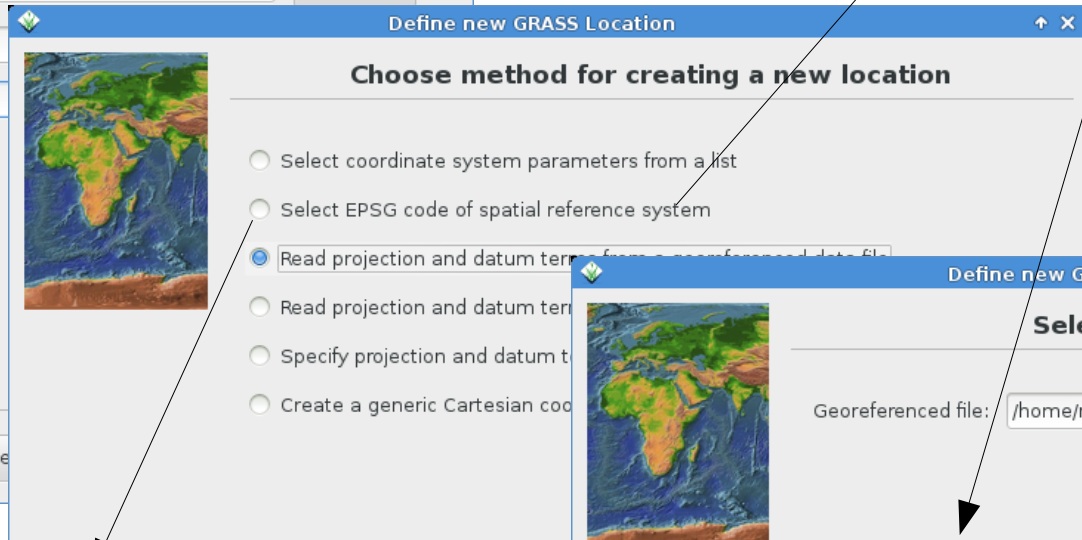
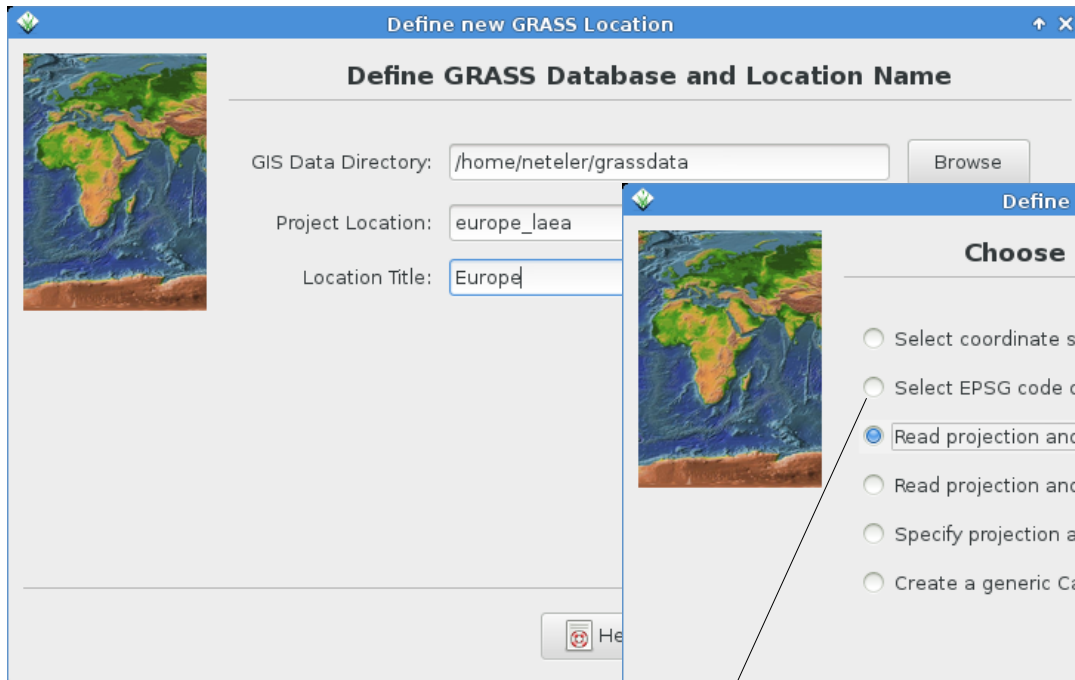
GRASS GIS 7 User interface



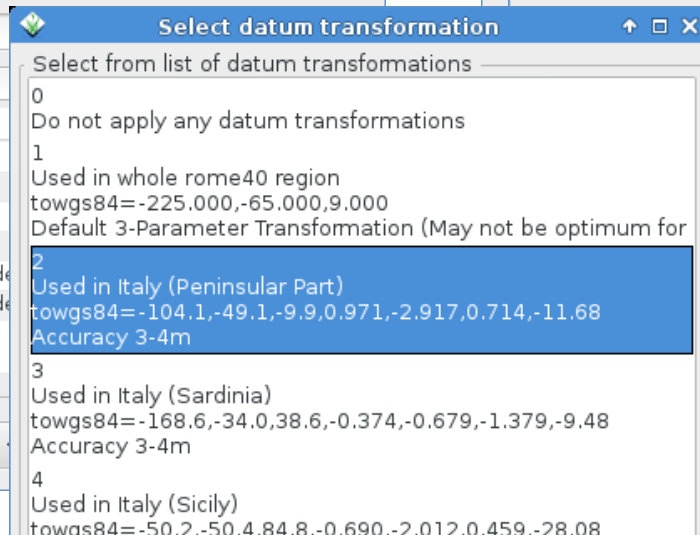
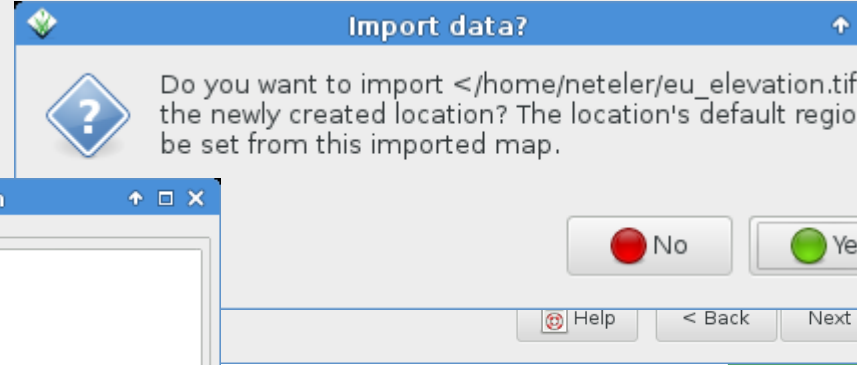
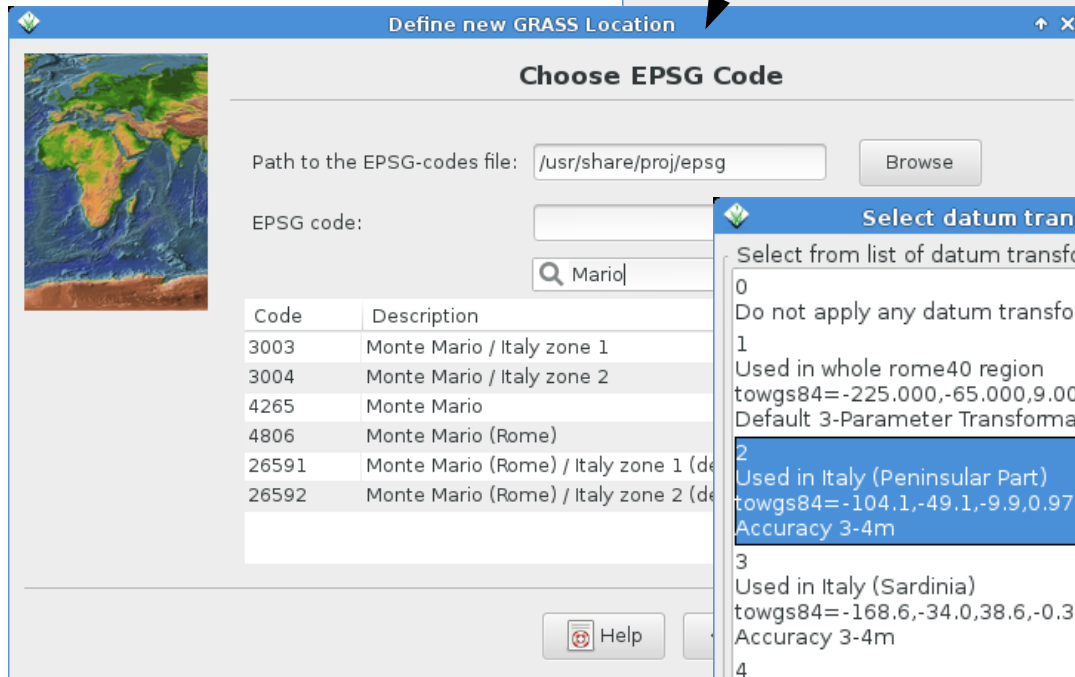
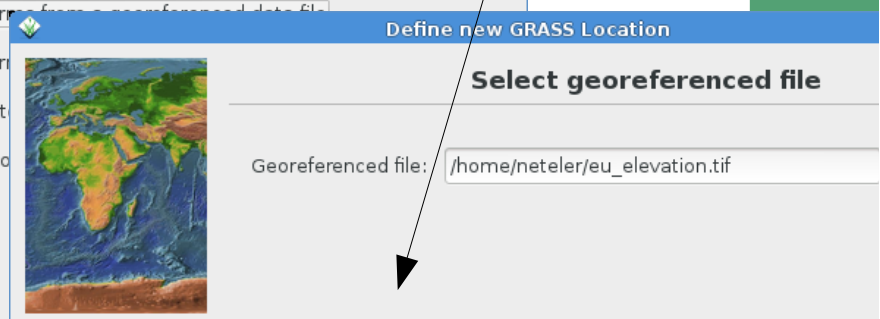
GRASS GIS: Interoperability



GRASS: Project database (Location) wizard



From GeoTIFF or SHAPE or ...
And directly import file



GRASS 7: New map histogram tool

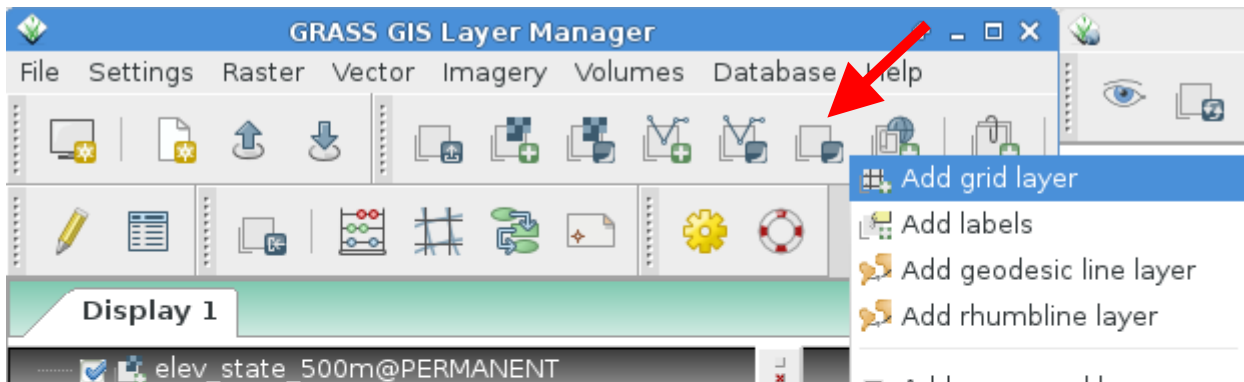
Using the Wake county "elevation" map:

The screenshot illustrates the workflow for creating a histogram in GRASS GIS. It shows three main windows:

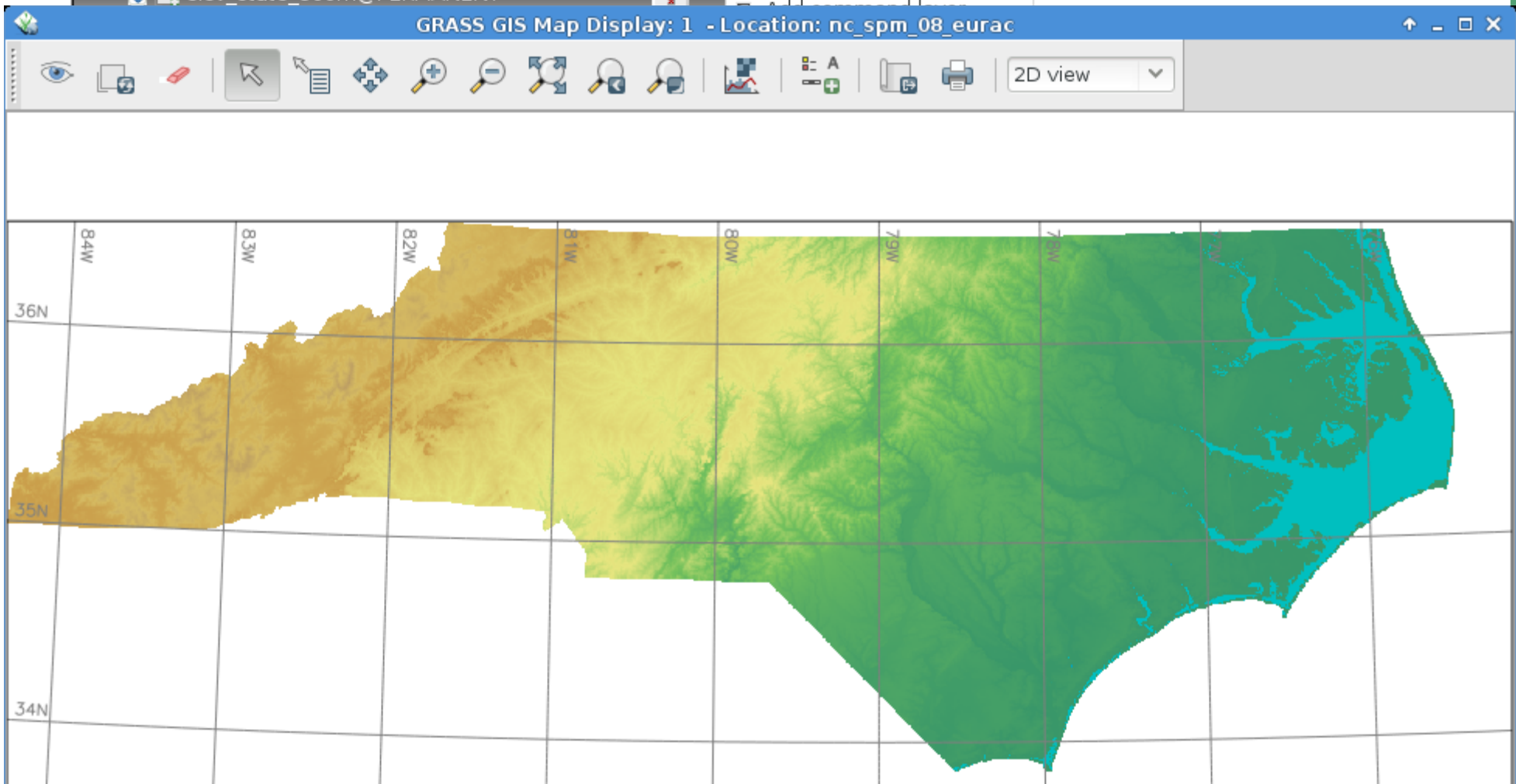
- GRASS GIS Map Display: 1 - Location: nc_spm_08_eurac**: The main map window displaying an elevation map. A red arrow points to the histogram icon in the toolbar.
- Select raster map or imagery group to histogram**: A dialog box where the user selects the raster map. The "elevation@PERMANENT" map is preselected in the "Select raster map:" field. Other options include "Histogram single raster" (selected), "Histogram imagery group", "Select image group:", "Number of bins (for FP maps)" (set to 255), and "Histogram type" (set to "count").
- GRASS Histogramming Tool**: The main histogram tool window showing a line graph titled "Histogram of elevation". The y-axis is labeled "Cell counts" (0 to 900) and the x-axis is labeled "Raster cell values" (60 to 160). The graph shows a distribution of cell counts across the elevation range.

Map will be preselected if selected in Layer Manager

GRASS 7: Adding a grid to the map view



- Grids
- Labels
- Geodesic lines
- Rhumbines



GRASS 7: New Geospatial Modeller

The screenshot displays the GRASS GIS Graphical Modeller window titled "GRASS GIS Graphical Modeller - gnip.gxm*". The main workspace shows a workflow diagram with the following components:

- Green boxes (Processes):** (1) db.select, (2) g.mapset, (3) v.in.ogr, (4) db.execute, (5) g.region, (7) r.mask, (9) v.surf.idw.
- Blue ovals (Variables):** rast mask%rok, output obs_%sloupec_%rok.
- Pink oval (Input/Output):** input/output vzorky.
- White rounded rectangle (Loop):** (6) sloupec in %sloupec.split(" ")
- White diamond (Decision):** (8) %method == 'idw'

The workflow starts with (1) db.select and (2) g.mapset leading to the input/output vzorky oval. (3) v.in.ogr also feeds into this oval. (4) db.execute feeds into (5) g.region. (5) g.region feeds into the loop (6). The loop (6) feeds into (7) r.mask. (7) r.mask feeds into the decision diamond (8). The decision diamond (8) has two paths: one leading to (9) v.surf.idw and another leading back to the loop (6). (9) v.surf.idw feeds into the output obs_%sloupec_%rok oval. The output oval also feeds back into (5) g.region.

An "if-else properties" dialog box is open in the foreground, showing the condition "%method == 'idw'". It lists items in the 'if' block and the 'else' block.

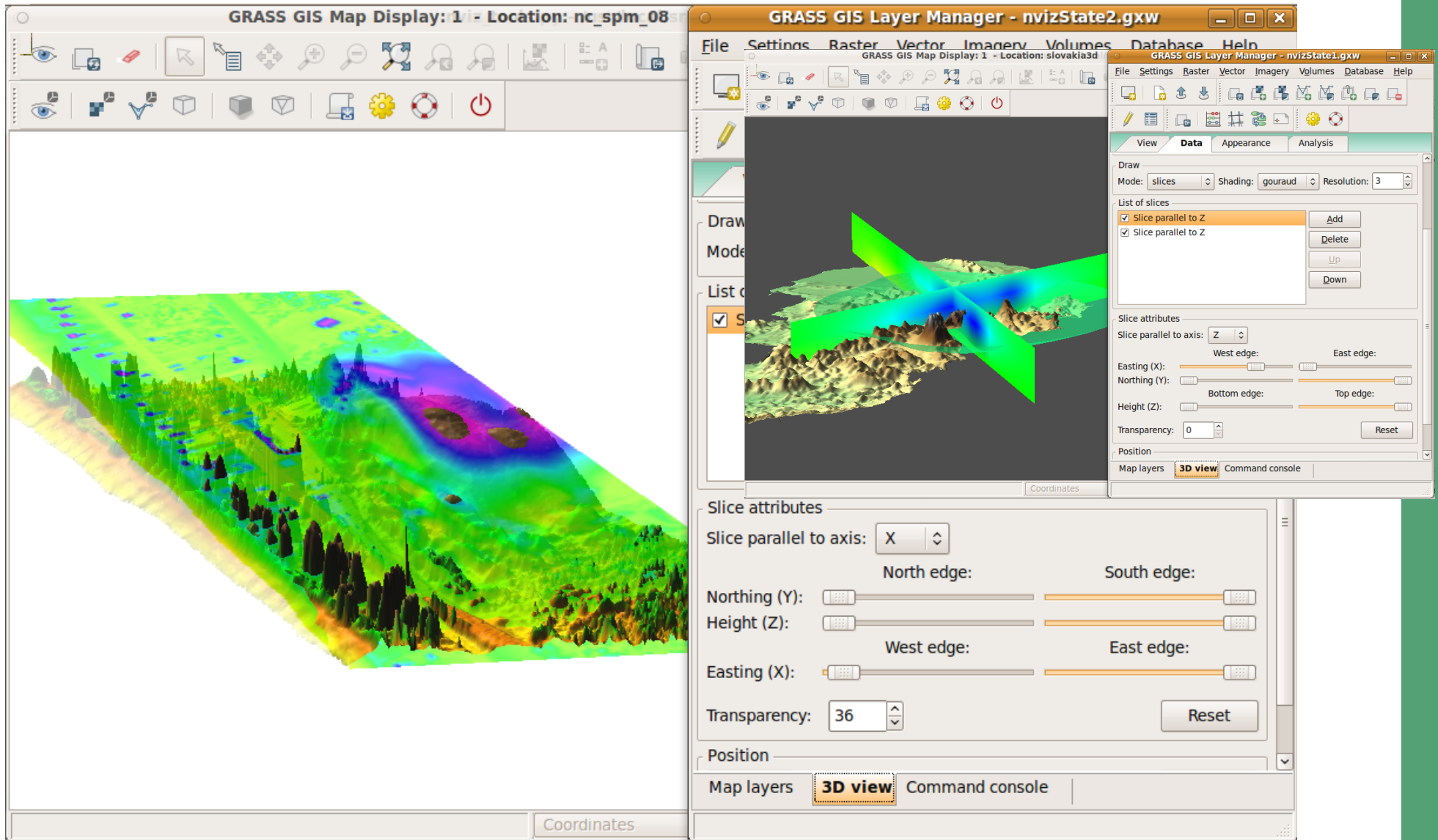
ID	Name	Command
<input type="checkbox"/>	4	db.execute input=/home/martin/grassdata/nc_...
<input type="checkbox"/>	5	g.region --overwrite rast=mask2006 res=10
<input type="checkbox"/>	6	loop Condition: sloupec in %sloupec.split(" ")
<input type="checkbox"/>	7	r.mask -r
<input checked="" type="checkbox"/>	9	v.surf.idw -n --overwrite input=vzorky output=o...

ID	Name	Command
<input type="checkbox"/>	1	db.select -c sql=SELECT id FROM farms WHER...
<input type="checkbox"/>	2	g.mapset -c mapset=vracov
<input type="checkbox"/>	3	v.in.ogr -o --overwrite dsn=PG:dbname=prefer...
<input type="checkbox"/>	4	db.execute input=/home/martin/grassdata/nc_...
<input type="checkbox"/>	5	g.region --overwrite rast=mask2006 res=10

At the bottom of the dialog, there are "Cancel" and "OK" buttons.

Below the screenshot, the text reads: *Extra bonus:*
export to Python scripts

GRASS 7: New visualization tool: wxNVIZ



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

Programming/screenshot:
Anna Kratochvílová

GRASS Topological 2D/3D Vector model

Vector geometry types

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

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

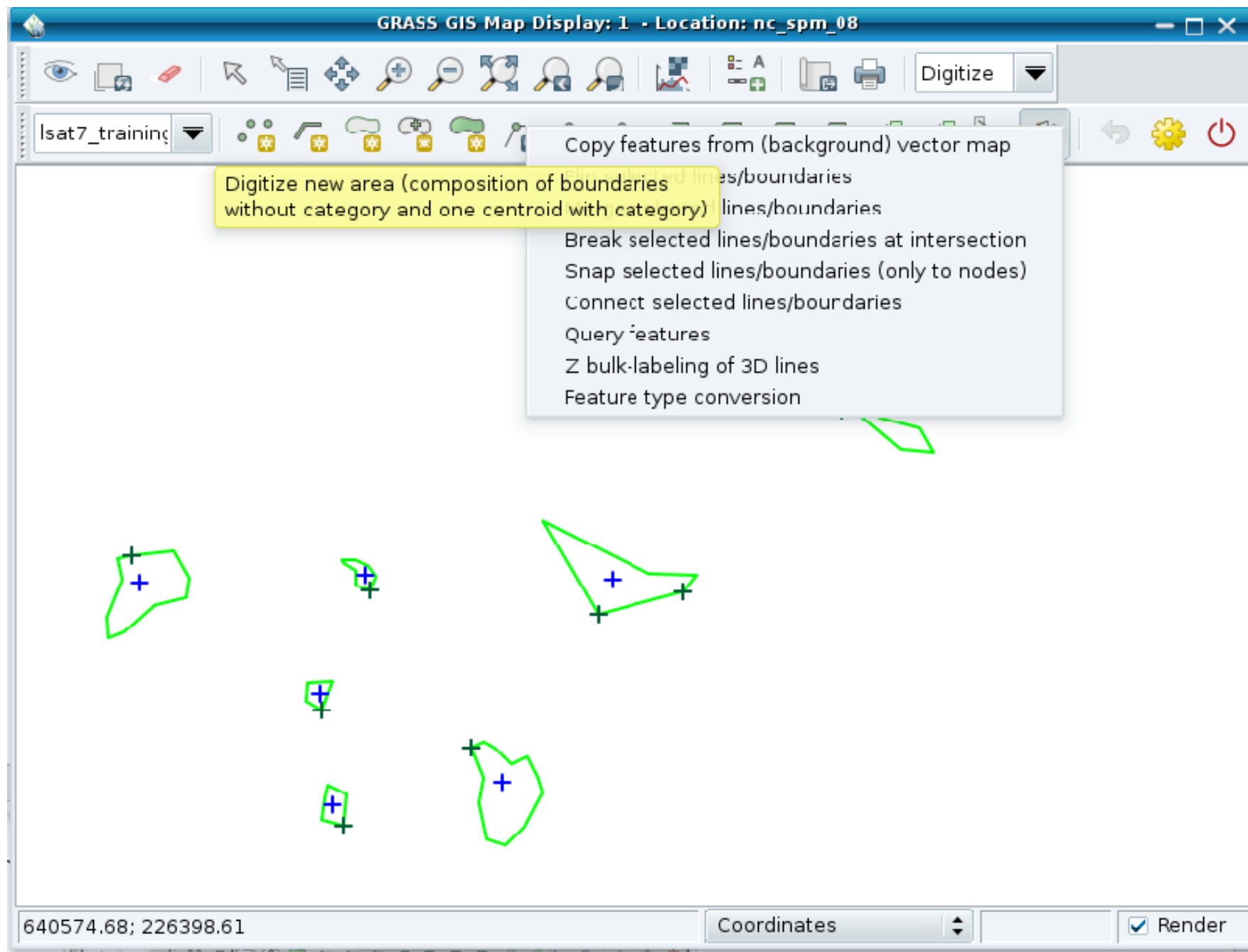


Faces

not in all GIS!



GRASS 7: Topological Vector Digitizer



GRASS 7: Topological Vector Digitizer in PostGIS 2 (under development)

Programmer: Martin Landa

The screenshot displays the GRASS GIS interface. On the left, the 'GRASS GIS Layer Manager' window shows the 'urbanar' layer selected. The 'Command console' at the bottom left shows the following information:

```
Map scale: 1:1
Name of creator: martin
Organization:
Source date: Thu Nov 29 13:30:15 2012
Timestamp (first layer): none
-----
Map format: PostGIS (PostgreSQL)
DB table: public.urbanarea
DB name: grass
Geometry column: geom
Feature type: polygon
Topology: PostGIS (schema: topo_urbanarea)
Topology column: topo
-----
Type of map: vector (level: 1)
```

The main window, 'GRASS GIS Map Display: 1 - Location: nc_spm_08', shows a map of urban areas with a digitizing toolbar at the top. An 'Update attributes' dialog box is open over the map, showing the following data for feature id 113:

Layer 1 / Category 113	Value
cat [integer]	55
objectid [integer]	55
ua [character]	x3261
name [character]	Raleigh
ua_type [character]	UA

The dialog box also includes a 'Feature id: 113' field, a checked 'Close dialog on submit' option, and 'Reload', 'Cancel', and 'Submit' buttons.

<http://grass.osgeo.org/grass70/manuals/v.out.postgis.html>

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

Cofunded by Municipality of Trento, Italy

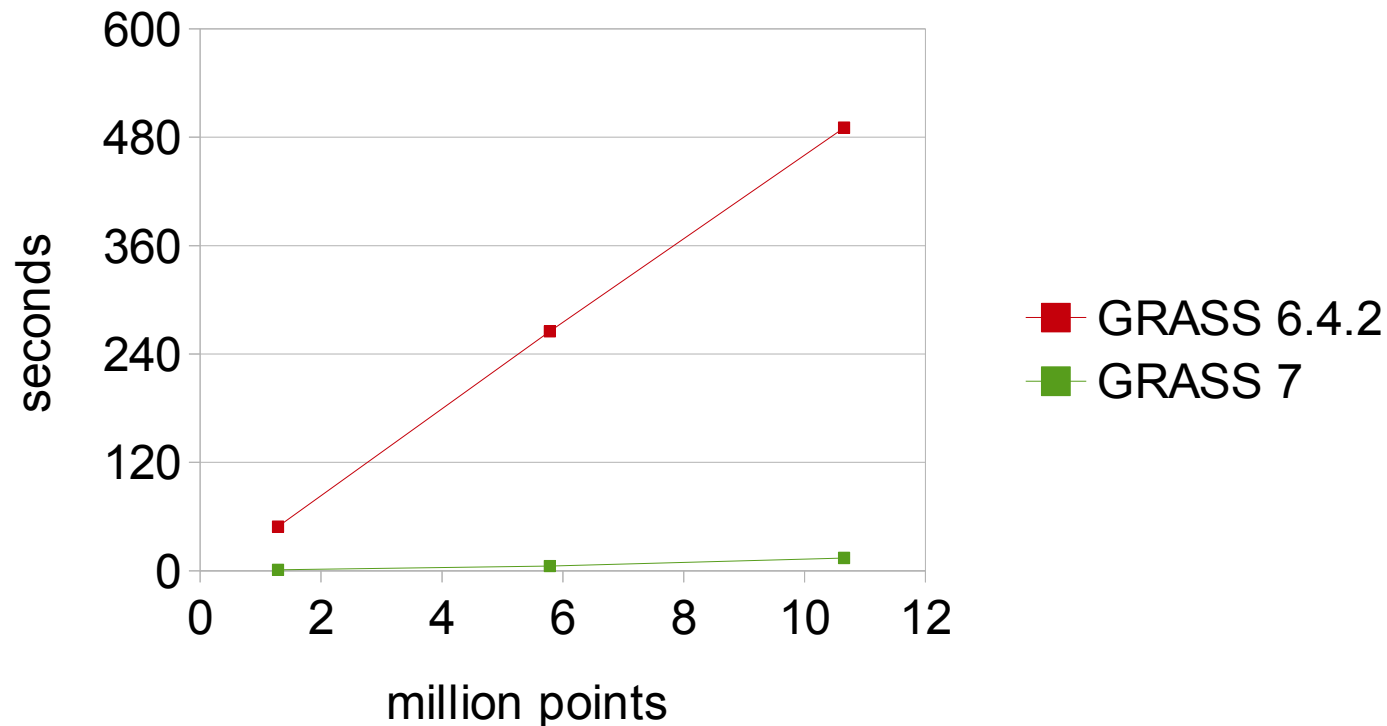
News in GRASS 7's Vector Topology

Spatial query example

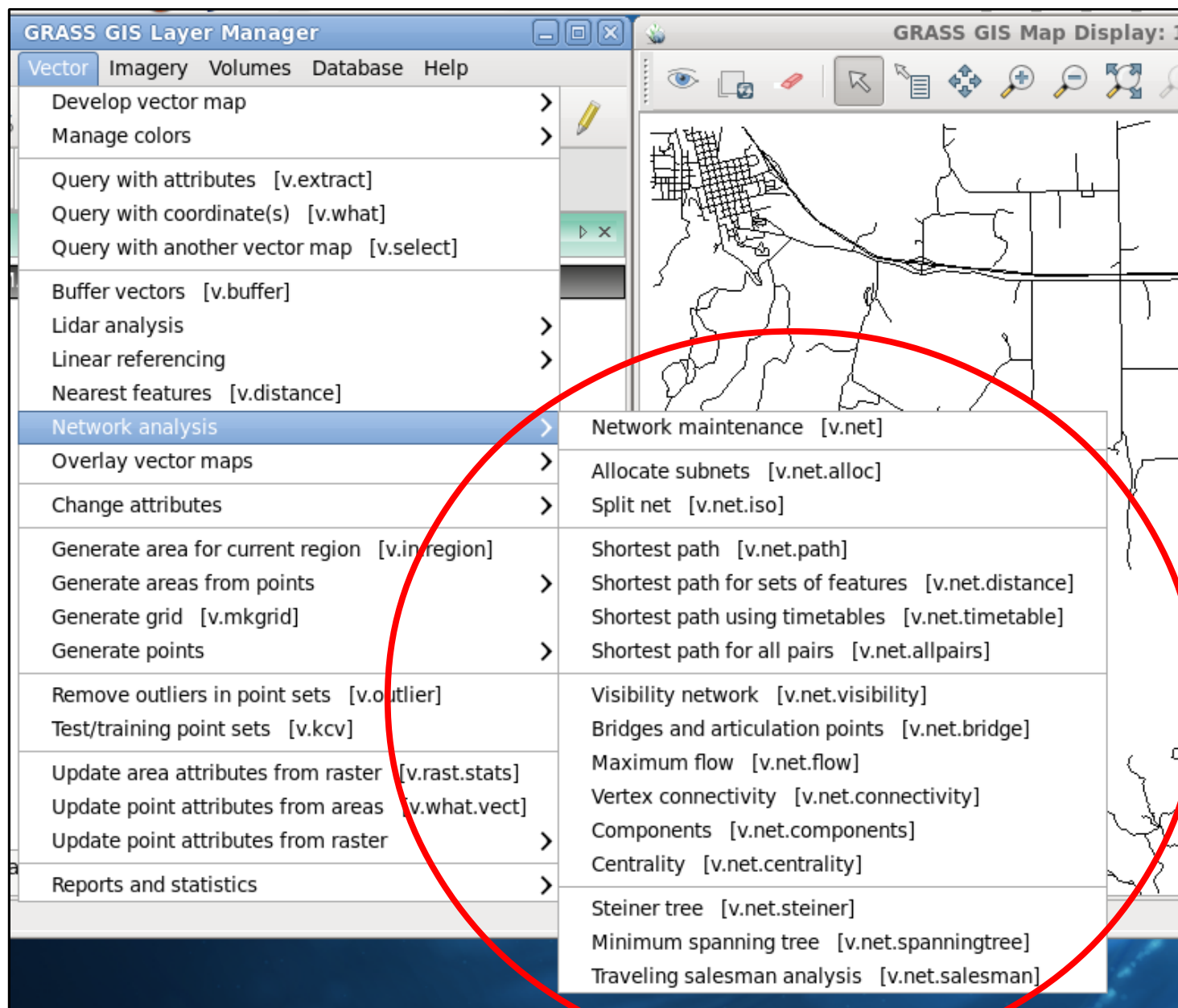
Query of vector point maps

GUI: click on vector map, what is there?

CLI: `v.what east_north=east,north`



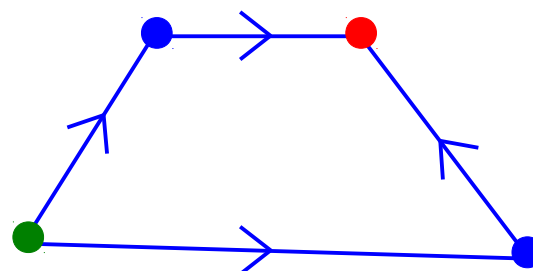
Vector network analysis in GRASS



Vector network analysis

General concept of a network graph

- Arcs connected by nodes
- Forward/backward costs assigned to each arc (oneway road)
- Starting point(s)
- Ending point(s)



Cost definition examples

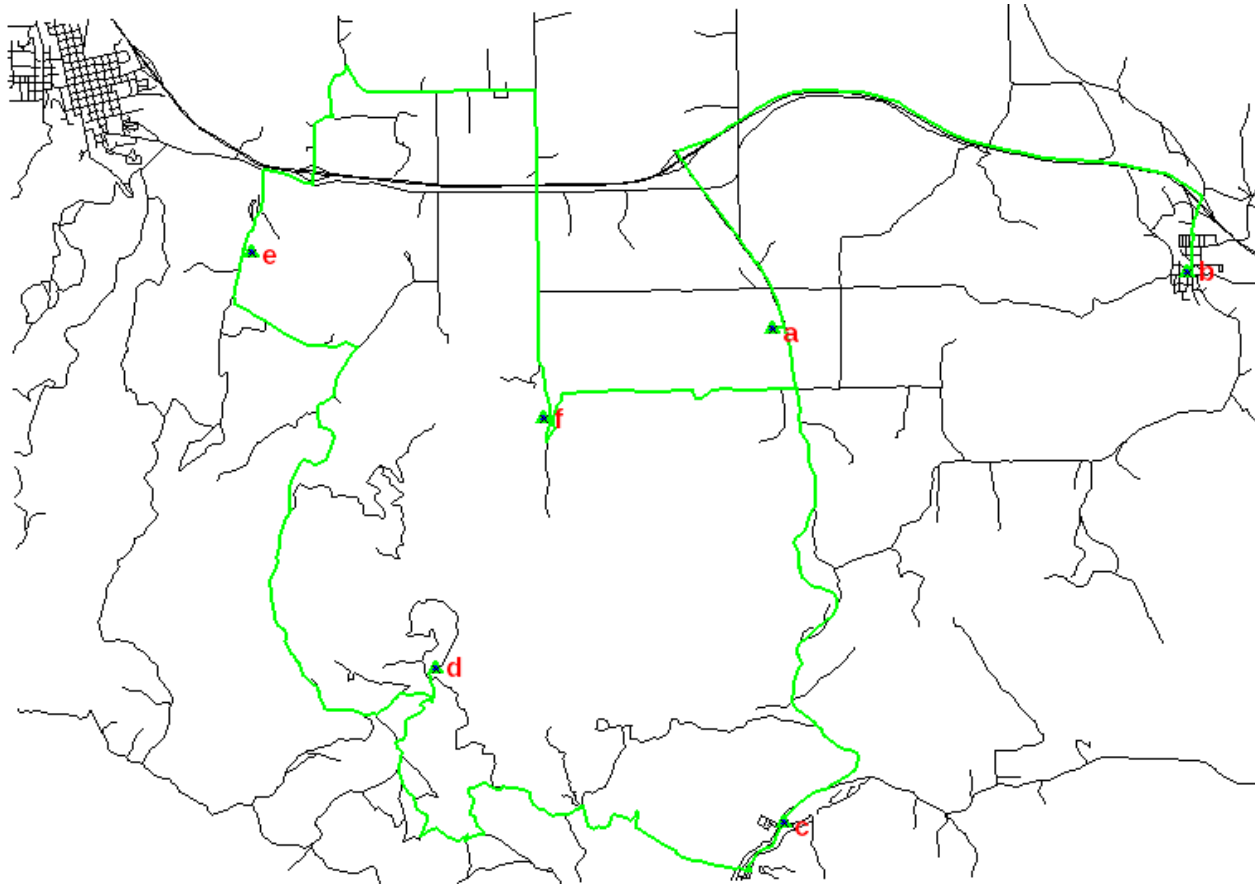
- Distance → shortest path
- Travelling time → fastest path
- Travelling costs (fuel, train ticket, etc) → cheapest path

GsoC 2013: implementation of turns support

Network analysis: traveling salesman

Traveling time as costs

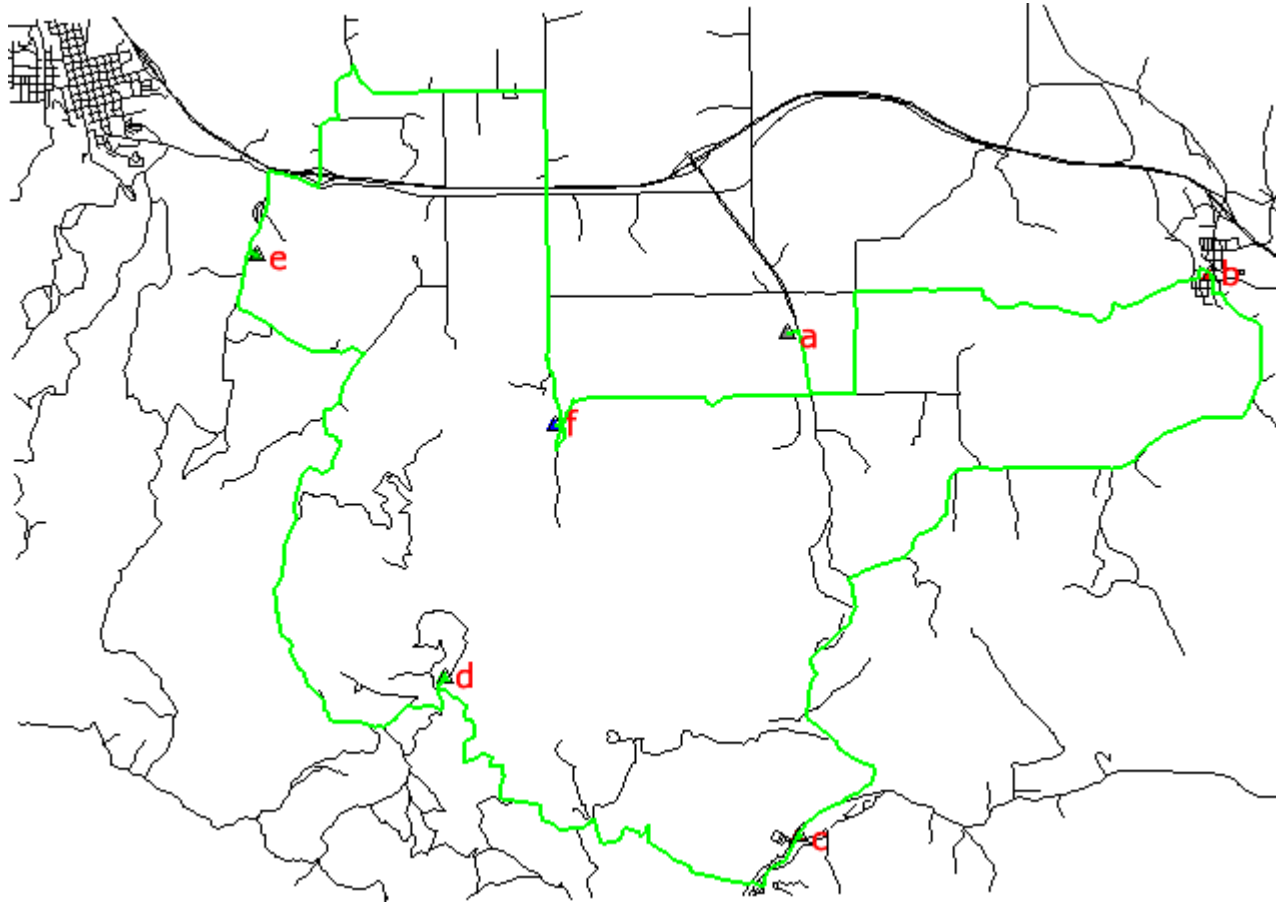
Result



Network analysis: traveling salesman

Distances as costs

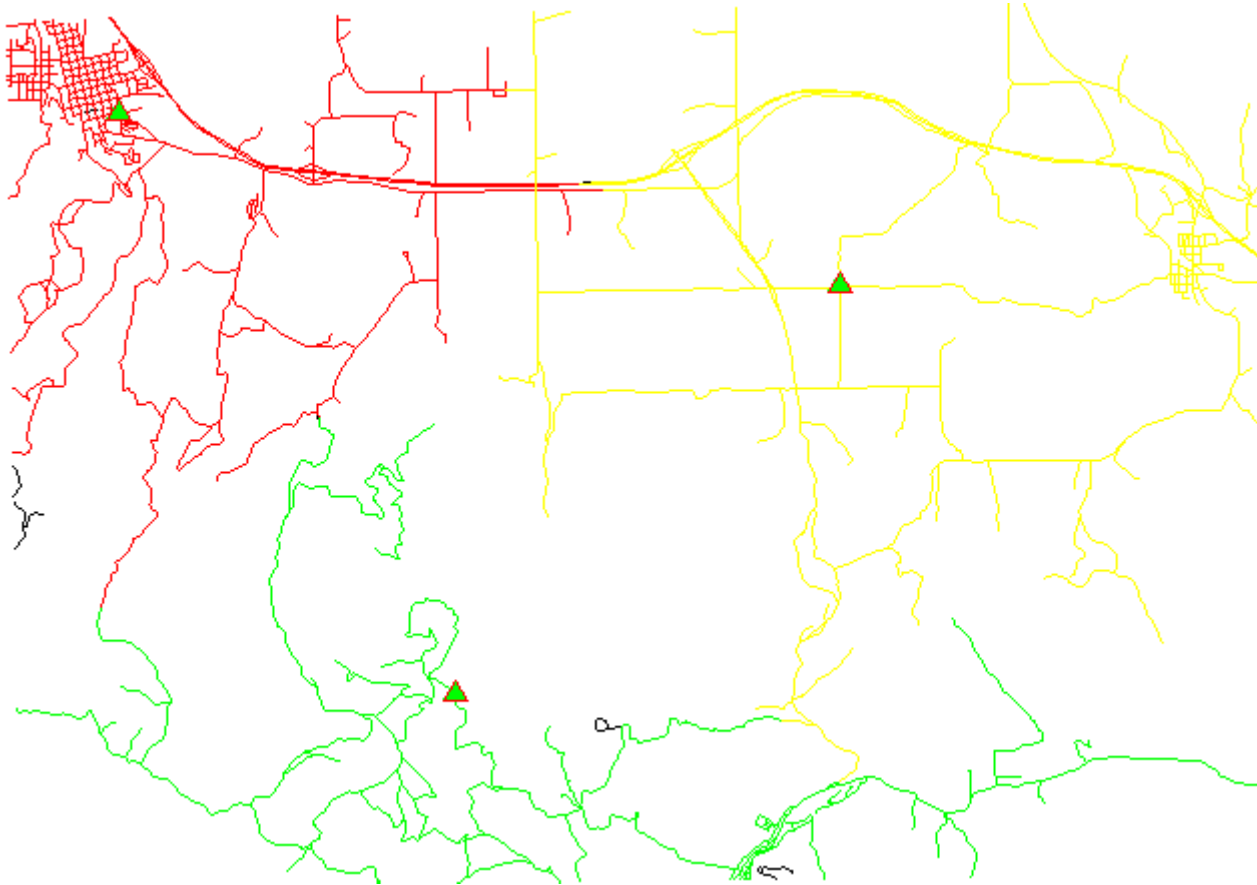
Result



Network analysis: allocate subnets

Distances as costs

Result



... and many more algorithms... see the manual!

New 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](#)

Statistics and gap filling

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

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](#)

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 7: New animation tool for time series

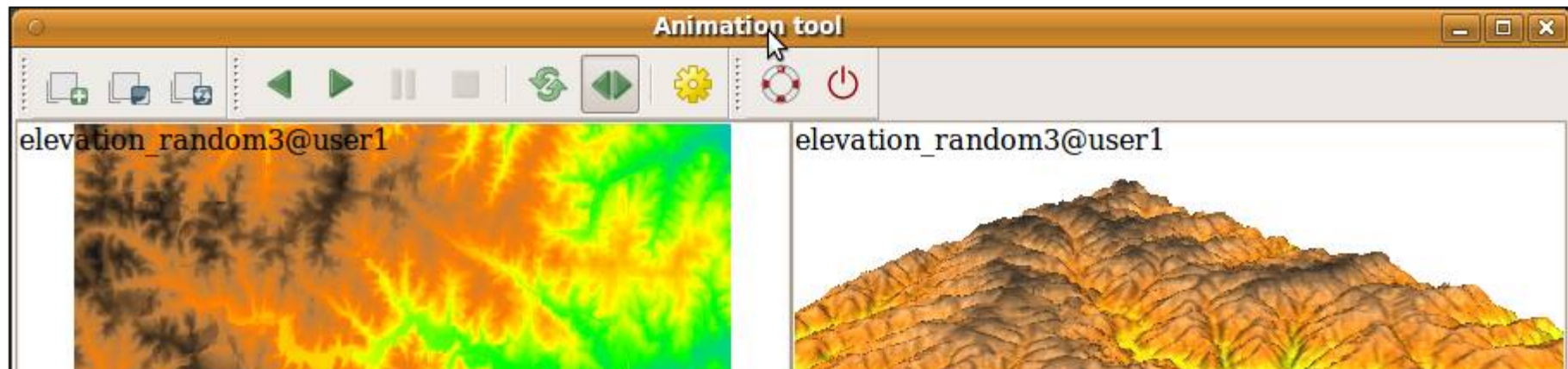
The **Animation Tool** is a [wxGUI](#) component for animating a series of GRASS raster maps or a space time raster dataset (created by t.* modules).

Animation Tool allows you to:

- display up to 4 synchronized animations
- control the animation speed
- interactively change active frame using a slider
- visualize space time datasets with unequally spaced intervals
- animate 3d view (partially implemented)

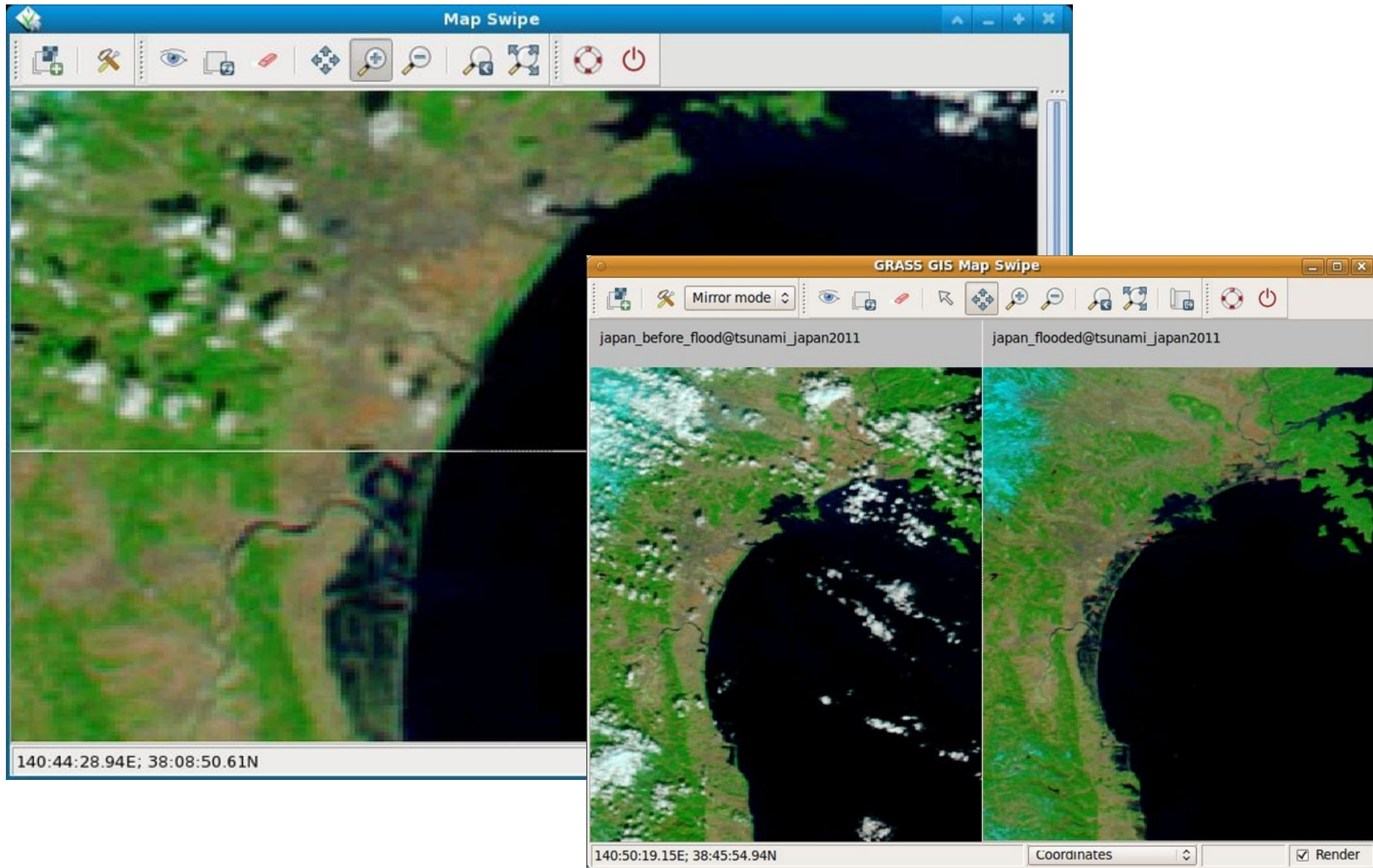
3D view animation enables to animate raster (as an elevation map or a color map) or vector map (points, lines). Internally, module m.nviz.image is used. To display 3D view animation follow these steps:

- open GRASS GUI, load maps and start 3D view
- set view, light and other parameters as you like
- save workspace file
- add new animation in Animation Tool, choose 3D view mode
- choose data (series of maps or space time dataset) used for animation
- set workspace file
- choose parameter (parameter of m.nviz.image) to animate (e.g. color_map)



<http://grass.osgeo.org/grass70/manuals/g.gui.animation.html>

New Map swiping tool for multitemporal maps

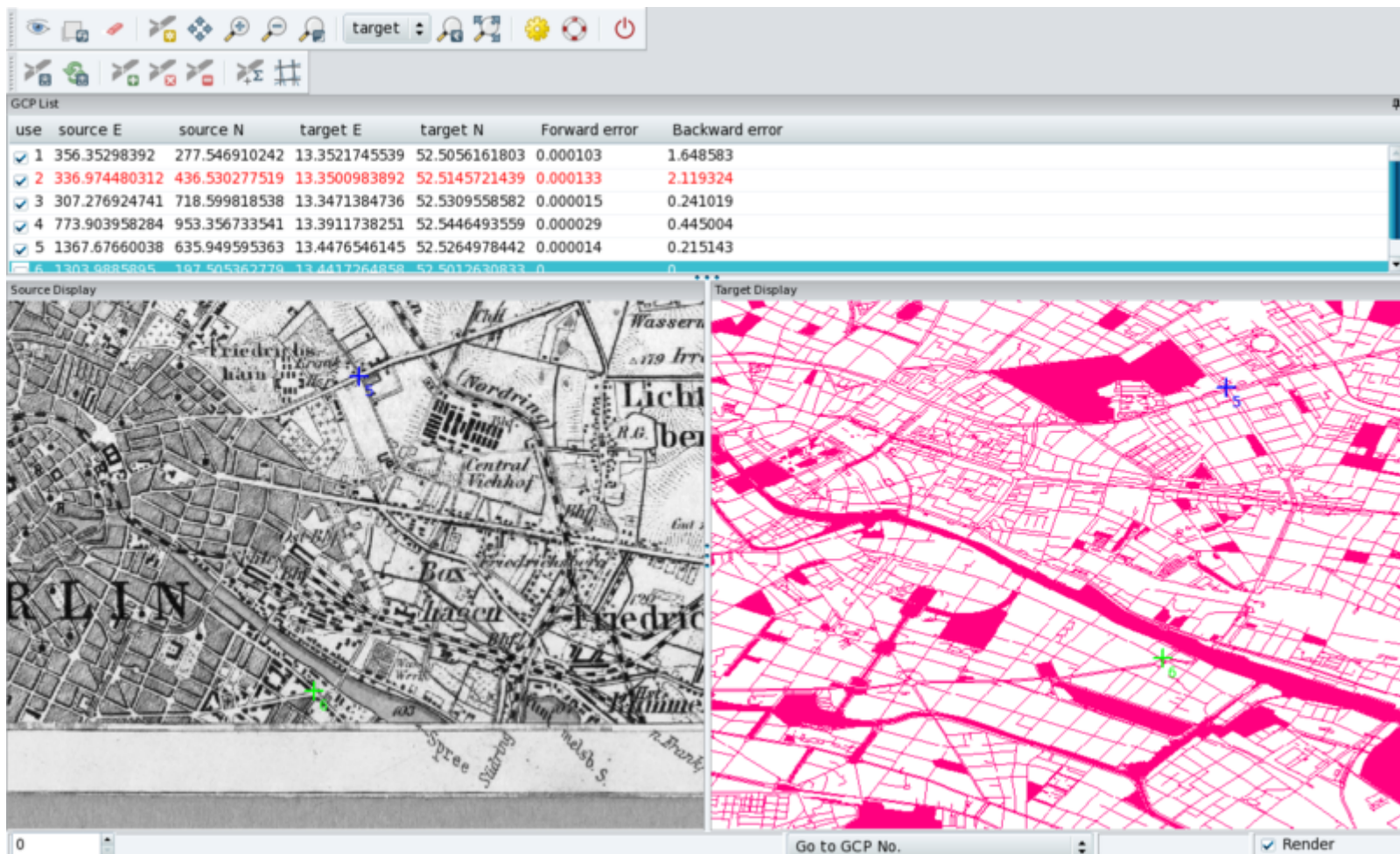
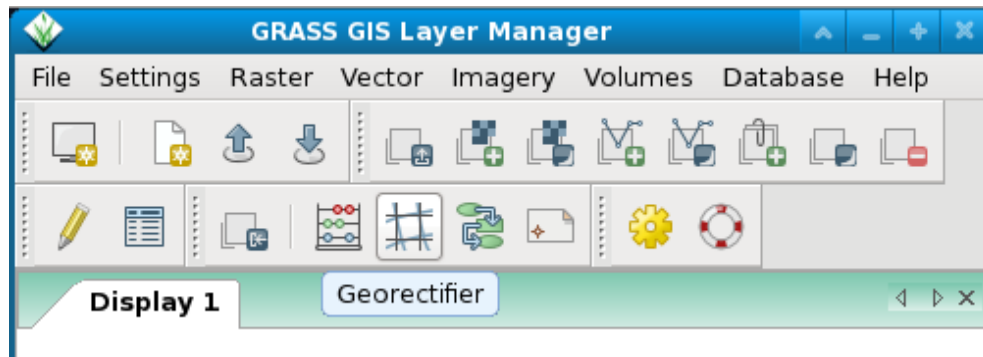


Pre and post disaster images of the tsunami in Japan in 2011
(MODIS images taken on February 26 and March 13, 2011)

GRASS 7: New geocoding tool

Image/Map rectifier

For raster (imagery, historic scans) and vector maps



use	source E	source N	target E	target N	Forward error	Backward error	
<input checked="" type="checkbox"/>	1	356.35298392	277.546910242	13.3521745539	52.5056161803	0.000103	1.648583
<input checked="" type="checkbox"/>	2	336.974480312	436.530277519	13.3500983892	52.5145721439	0.000133	2.119324
<input checked="" type="checkbox"/>	3	307.276924741	718.599818538	13.3471384736	52.5309558582	0.000015	0.241019
<input checked="" type="checkbox"/>	4	773.903958284	953.356733541	13.3911738251	52.5446493559	0.000029	0.445004
<input checked="" type="checkbox"/>	5	1367.67660038	635.949595363	13.4476546145	52.5264978442	0.000014	0.215143
<input type="checkbox"/>	6	1303.9885895	107.505362779	13.4417264858	52.5012630833	0	0

GRASS 7 Addons: User contributed extensions

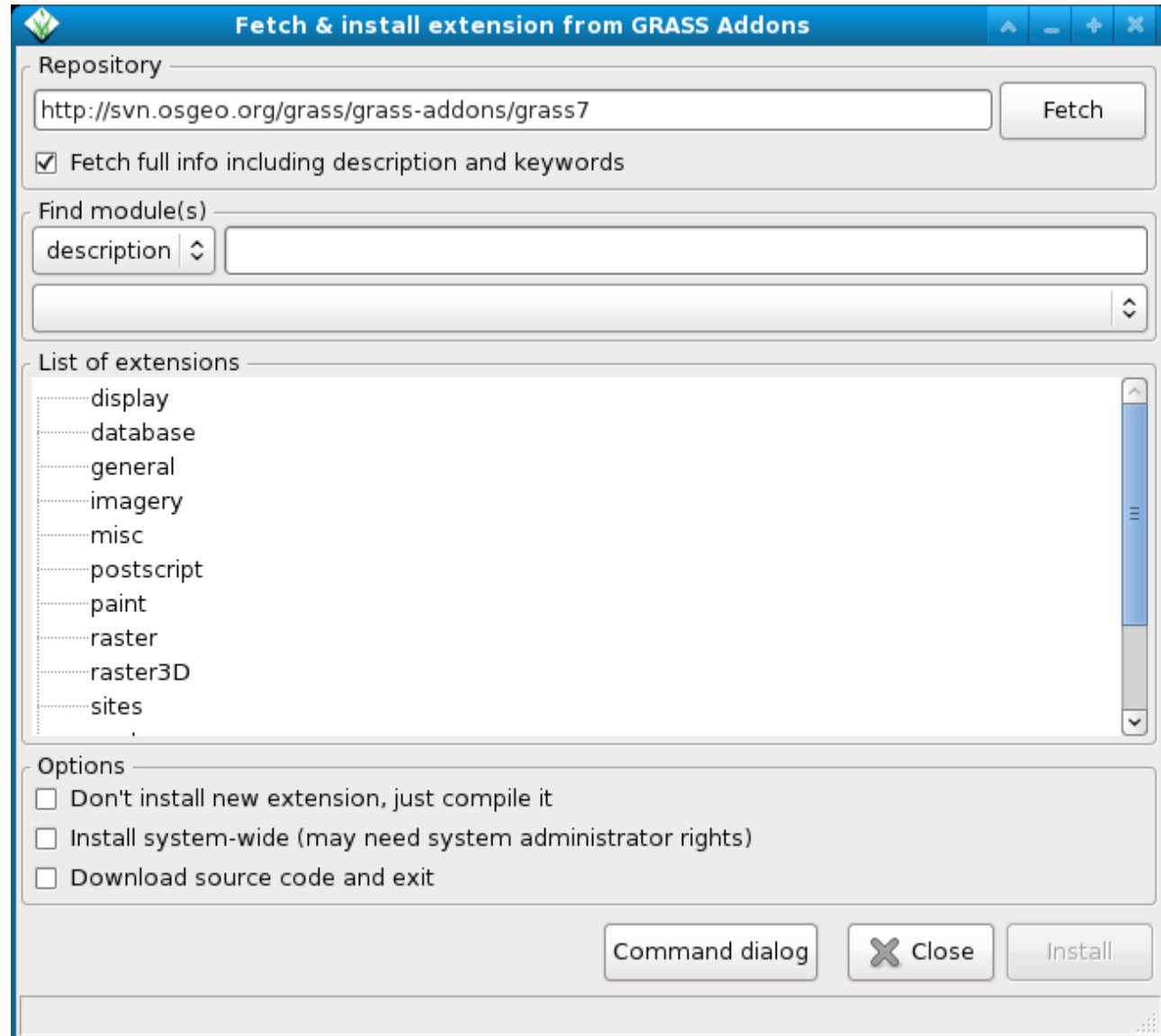
The Addons repository is SVN based:

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

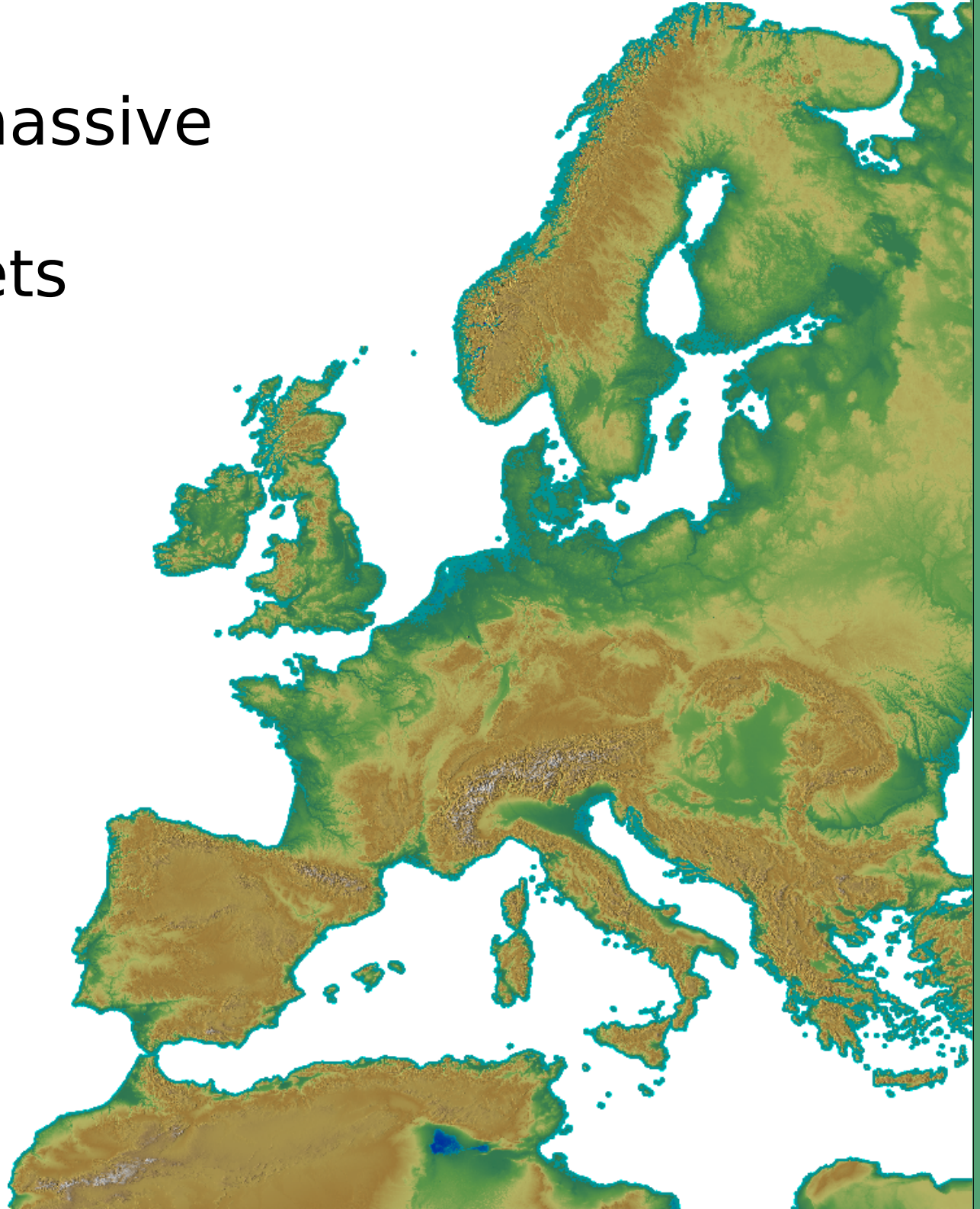
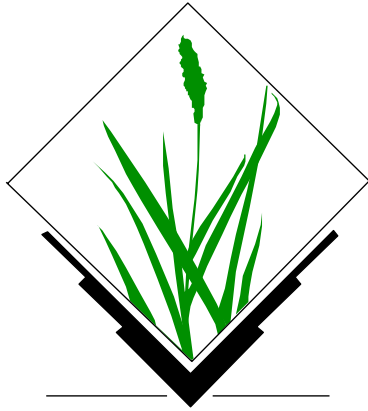
<http://grass.osgeo.org/wiki/Addons>

Installation with
extension manager

Increasing inflow
of Python scripts



Support for massive spatial datasets in GRASS GIS



GRASS 7: Support for massive datasets

What is massive?





Massive is relative to

Hardware resources

Software capabilities

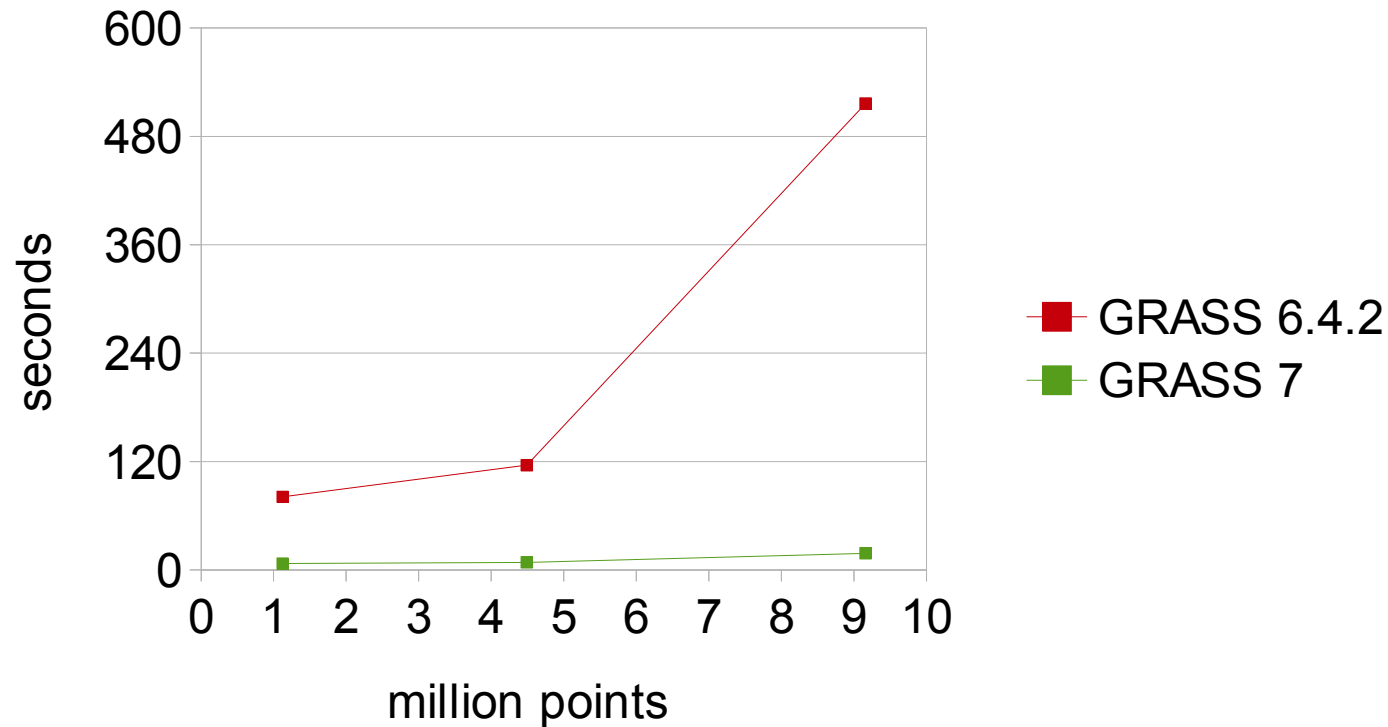
Operating system capabilities

Limiting factors

-  RAM
-  Processing time
-  Disk space
-  Largest supported file size

GRASS 7: Support for massive datasets

Cost surfaces: *r.cost*



Other speed figure:
**PCA of 30 million pixels
in 6 seconds** on this small
presentation laptop...

New tools for hydrological modelling

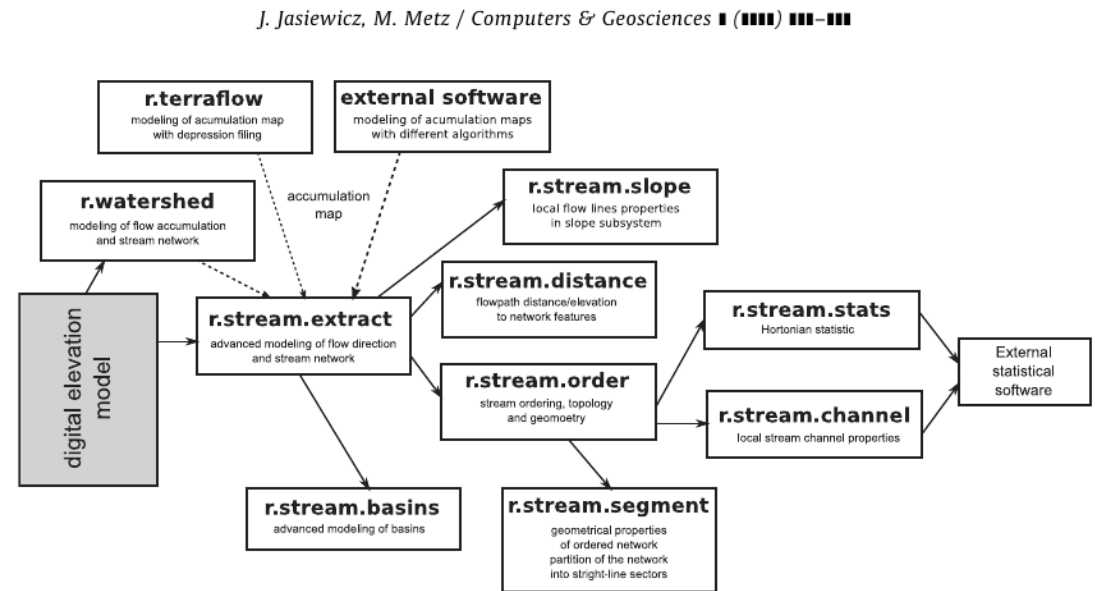
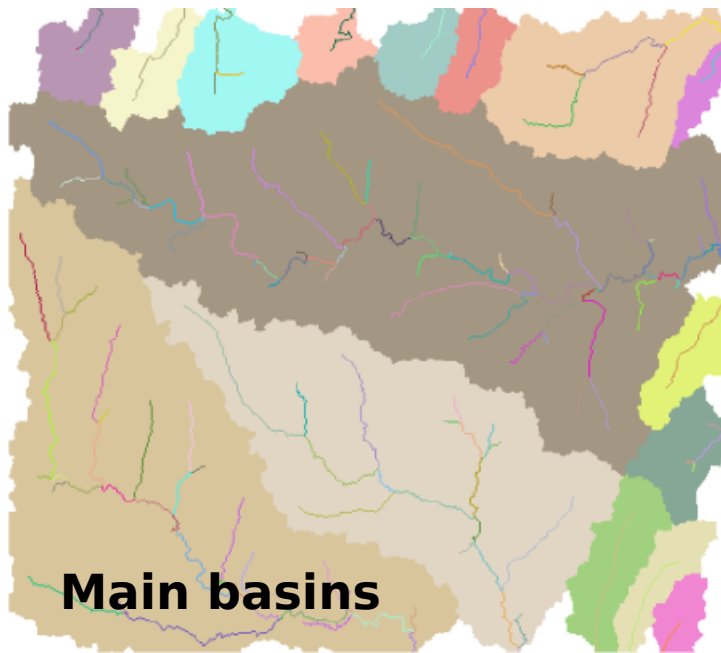


Fig. 2. The structure of the r.stream toolset and data flow between particular modules and external software.



A new GRASS GIS toolkit for Hortonian analysis of drainage networks

Jarosław Jasiewicz^{a,*}, Markus Metz^b

^a Adam Mickiewicz University, Geology and Geoinformation Institute, Działgiewna 27 60-081 Poznań, Poland
^b University of Osnabrück, Institute of Experimental Ecology, Allee 11, 49069 Osnabrück, Germany

ARTICLE INFO

Article history:
Received 8 March 2010
Received in revised form
24 February 2011
Accepted 2 March 2011

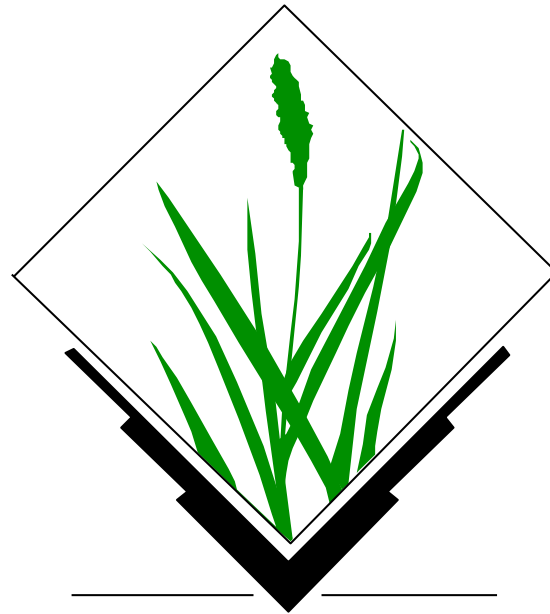
Keywords:
Drainage network
Multiple flow direction
Basis delineation
GRASS GIS
Network topology

ABSTRACT

The aim of this paper is to present a new GRASS GIS toolkit designed for Hortonian analysis of drainage networks. The r.stream toolkit uses a multiple flow direction algorithm for stream network extraction as well as for calculating other hydrogeomorphological features in the catchment's area. As all GRASS GIS toolsets, r.stream consists of several separate modules that can extract stream networks from a spectrum of accumulation maps, order the extracted network using several ordering methods, do advanced modeling of basin's boundary, perform Hortonian statistics, calculate additional parameters such as flow path distance to watershed elements, partition ordered and unordered networks into near-straight-line sectors, and calculate sector directions. The package is free and open-source software, available for GRASS version 6.4 and later.

© 2011 Elsevier Ltd. All rights reserved.

Programming own applications with GRASS GIS 7



GRASS and Python

Parser part (for GUI autocreation and command line support):

```
#!/Module
#% description: Drapes a color raster over a shaded relief map using d.his
#%End
#%option
#% key: reliefmap
#% type: string
#% gisprompt: old,cell,raster
#% description: Name of shaded relief or aspect map
#% required : yes
#%end
#%option
#% key: drapemap
#% type: string
#% gisprompt: old,cell,raster
#% description: Name of raster to drape over relief map
#% required : yes
#%end
```

Script part:

```
import sys
from grass.script import core as grass

def main():
    drape_map = options['drapemap']
    relief_map = options['reliefmap']
    brighten = options['brighten']
    ret = grass.run_command("d.his", h_map = drape_map, i_map = relief_map, brighten = brighten)
    sys.exit(ret)

if __name__ == "__main__":
    options, flags = grass.parser()
    main()
```

New GRASS 7 Python API

 [Neteler](#) [My talk](#) [My preferences](#) [My watchlist](#) [My contributions](#) [Log out](#)



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](#)

Contents [\[hide\]](#)

- 1 [Python SIGs](#)
- 2 [Writing Python scripts in GRASS](#)
 - 2.1 [Python script editor](#)
 - 2.2 [Using the GRASS Python Scripting Library](#)
 - 2.3 [Creating Python scripts](#)
 - 2.3.1 [MS-Windows](#)
 - 2.3.2 [Linux](#)
 - 2.4 [Running external commands](#)
 - 2.5 [Testing and installing scripts](#)
 - 2.5.1 [Debugging](#)
 - 2.5.2 [Installation](#)
- 3 [Python extensions in GRASS](#)
 - 3.1 [Python Scripting Library](#)
 - 3.2 [Python Ctypes Interface](#)
 - 3.3 [wxPython GUI development](#)
 - 3.4 [Python-GRASS add-on](#)
 - 3.5 [Using GRASS GUI from Python](#)
- 4 [FAQ](#)
- 5 [Links](#)
 - 5.1 [General guides](#)
 - 5.2 [Programming](#)
 - 5.3 [Presentations](#)

Table Of Contents

[Welcome to PyGrass's documentation!](#)
[Indices and tables](#)

Next topic

[Raster](#)

This Page

[Show Source](#)

Quick search

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

Welcome to PyGrass's documentation!

Since in the 2006 GRASS developers start to adopt python for the new GUI, python becoming more and more important and developers plan to convert all the bash scripts in to python for the next major release GRASS 7.

`pygrass` want to improve integration between GRASS and python, make the use of python under GRASS more consistent with the language itself and make the GRASS scripting and programming activity easier and more natural to the final users.

This project has been funded with support from the google Summer of Code 2012.

Contents:

- [Raster](#)
 - [Categories](#)
 - [RastRow](#)
 - [RasterRowIO](#)
 - [RastSegment](#)
 - [RasterNumpy](#)
 - [Buffer](#)
 - [RowIO](#)
 - [Segment](#)
 - [History](#)
 - [Category](#)
- [Vector](#)
 - [VectorTopo](#)
 - [Vector](#)

New GRASS 7 Python API

ISPRS Int. J. Geo-Inf. **2013**, *2*, 201-219; doi:10.3390/ijgi2010201

OPEN ACCESS

ISPRS International
Journal of
Geo-Information
ISSN 2220-9964
www.mdpi.com/journal/ijgi

Article

Pygrass: An Object Oriented Python Application Programming Interface (API) for Geographic Resources Analysis Support System (GRASS) Geographic Information System (GIS)

Pietro Zambelli^{1,*}, **Sören Gebbert**² and **Marco Ciolli**¹

¹ Department of Civil, Environmental and Mechanical Engineering, University of Trento, via Mesiano 77, Trento 38123, Italy; E-Mail: marco.ciolli@ing.unitn.it

² Thünen Institute of Climate-Smart Agriculture, Bundesallee 50, Braunschweig 38116, Germany; E-Mail: soerengebbert@googlemail.com

* Author to whom correspondence should be addressed; E-Mail: pietro.zambelli@ing.unitn.it; Tel.: +39-46-128-2696.

Received: 1 January 2013; in revised form: 21 January 2013 / Accepted: 21 February 2013 /

Published: 11 March 2013

Abstract: PyGRASS is an object-oriented Python Application Programming Interface (API) for Geographic Resources Analysis Support System (GRASS) Geographic Information System (GIS), a powerful open source GIS widely used in academia, commercial settings and governmental agencies. We present the architecture of the PyGRASS library, covering interfaces to GRASS modules, vector and raster data, with a focus on the new capabilities that it provides to GRASS users and developers. Our

<http://dx.doi.org/10.3390/ijgi2010201>

GRASS 7 Programmer's manual

grass.osgeo.org/programming7/

GRASS Programmer's Manual 7.0.svn(2013)-r56710

Main Page Related Pages Namespaces Data Structures Files Search

GRASS 7 Programmer's Manual

GRASS GIS (**Geographic Resources Analysis Support System**) is an open source, free software *Geographical Information System* (GIS) with raster, topological vector, image processing, and graphics production functionality that operates on various platforms through a graphical user interface (GUI) or command line interface (CLI). It is released under [GNU General Public License \(GPL\)](#).

This manual introduces the reader to the *Geographic Resources Analysis Support System* from the programming perspective. Design theory, system support libraries, system maintenance, and system enhancement are all presented. This work is part of ongoing research being performed by the [GRASS Development Team](#), an international team of programmers, GRASS module authors are cited within their module's source code and the contributed manual pages.

© 2000-2013 by the GRASS Development Team

This manual is published under [GNU Free Documentation License \(GFDL\)](#), and comes with **ABSOLUTELY NO WARRANTY**. The development of GRASS software and this manual is kindly supported by the [Open Source Geospatial Foundation](#), who provides the GRASS main infrastructure.

Main web site: <http://grass.osgeo.org>

Note: Missing entries below still need to be documented in Doxygen format.

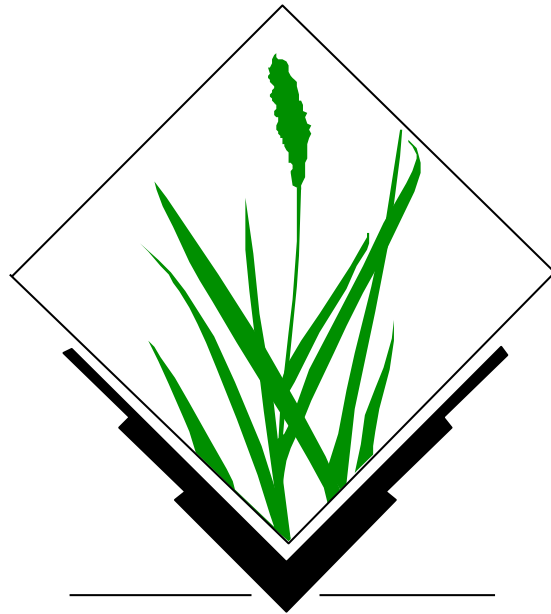
grass.osgeo.org/programming7/pages.html

Generated on Sat Jun 15 2013 00:38:34 for GRASS Programmer's Manual by [doxygen](#) 1.8.3

<http://grass.osgeo.org/programming7/>

GRASS GIS as Open Source GIS backbone:

Connecting to other software packages



New GRASS and QGIS Integration: Sextante



A screenshot of the QGIS 1.8.0-Lisboa software interface. The main window shows a map of Italy with a dark red vector layer. The 'Analysis' menu is open, displaying the 'SEXTANTE Toolbox' option. On the right side, the 'SEXTANTE Toolbox' panel is visible, listing various processing tools under the 'Vector (v.*)' category, with 'v.generalize - Vector based generalization' selected. In the foreground, the 'v.generalize - Vector based generalization' dialog box is open, showing parameters for the tool: 'Name of input vector map' is 'italy_wgs84', 'method' is 'douglas', 'Maximal tolerance value' is '1.0', and 'Look-ahead parameter' is '7'. The progress bar at the bottom of the dialog shows 0% completion.

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

SEXTANTE – GRASS Integration: Modeller

SEXTANTE



The screenshot displays the SEXTANTE Modeller interface. The main window, titled "Modeller", shows a workflow diagram for "Watershed modelling" under the group "Calculus tools for raster layer". The workflow consists of the following steps:

- DEM (input)
- r.fillnulls (intermediate step)
- r.contour (intermediate step)
- r.fill.dir (intermediate step)
- r.watershed (basin) (intermediate step)
- Vectorize raster layer (polygons) (final step)

A "Parameters" dialog box for "Watershed modelling" is open in the foreground, showing the following settings:

- Parameters: Raster output
- Inputs: Raster layers (DEM: wake_elevation.tif)
- Outputs: contour_lines10m[vector] (/home/markus/contours_10m.shp), basins[vector] (/home/markus/wake_watersheds.shp)

Buttons for "Run", "New", "Save", "Open", "OK", "Cancel", and "i" are visible.

Note:
Partially functional in
gvSIG OADE 2010,
more to come in (near) future...

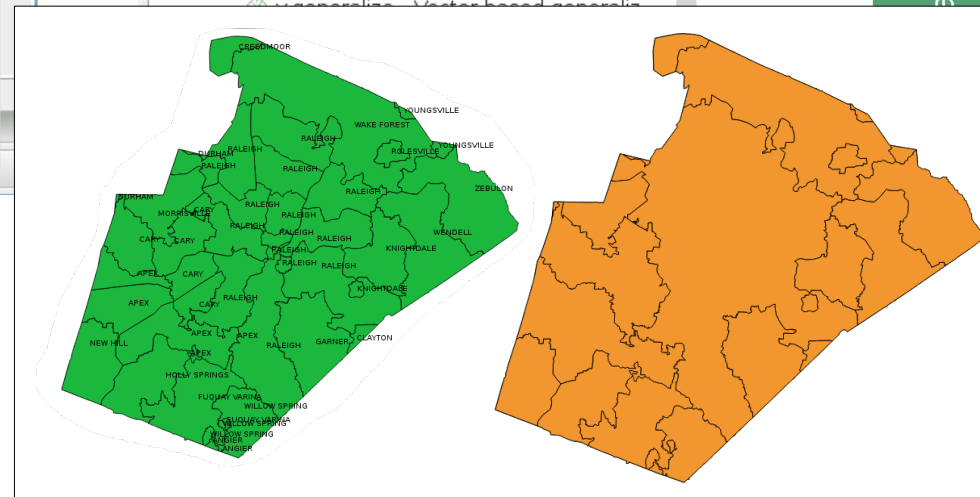
QGIS-Sextante – Using GRASS in QGIS via Sextante

The image shows a composite screenshot of the QGIS software interface. At the top left, the 'Vector' menu is open, and the 'Add Vector Layer' option is highlighted with a red arrow and a green circle labeled '1'. In the center, the 'Add vector layer' dialog box is open, showing the 'File' source type selected, the encoding set to 'System', and the dataset path 'course2013/data/gis_data/zipcodes_wake.shp'. A green circle labeled '2' is placed over the dialog's title bar. At the bottom right, the 'SEXTANTE Toolbox' is visible, with a red arrow pointing to the 'GRASS commands' entry, which is highlighted with a green circle labeled '3'. The main map area shows a green-shaded map with a black outline of a geographic region. The 'Layers' panel on the left shows the 'zipcodes_wake' layer loaded.

QGIS-Sextante – Vector dissolving example

The screenshot shows the QGIS 1.8.0-Lisboa interface. The main window displays a map with a green polygon. The 'v.dissolve' tool dialog is open, showing the 'Parameters' tab. The 'Input vector layer' is set to 'zipcodes_wake'. The 'Name of column used to dissolve common boundaries' list includes 'cat', 'NAME', 'OBJECTID', 'PERIMETER', 'SHAPE_Area', 'SHAPE_Leng', 'WAKE_ZIPCO', 'ZIPCODE', 'ZIPCODE_', 'ZIPCODE_ID', 'ZIPNAME', and 'ZIPNUM'. The 'ZIPNAME' option is selected. The 'Sextante Toolbox' is also visible, showing a list of algorithms under the 'Vector (v.*)' category, with 'v.dissolve - Dissolves boundaries bet...' highlighted.

Dissolving geometry by string column attributes:
Sextante calls GRASS GIS in a virtual session which deliver the result back (here SHAPE file)



GRASS 7: Native WPS Support

```
r.grow --wps-process-description
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd"
service="WPS" version="1.0.0" xml:lang="en-US">
  <ProcessDescription wps:processVersion="1" storeSupported="true" statusSupported="true">
    <ows:Identifier>r.grow</ows:Identifier>
    <ows:Title>Generates a raster map layer with contiguous areas grown by one cell.</ows:Title>
    <ows:Abstract>The manual page of this module is available here: http://grass.osgeo.org/grass70/manuals/html70\_use</ows:Abstract>
    <ows:Metadata xlink:title="raster" />
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>input</ows:Identifier>
        <ows:Title>Name of input raster map</ows:Title>
        <ComplexData maximumMegabytes="2048">
          <Default>
            <Format>
              <MimeType>image/tiff</MimeType>
            </Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>image/tiff</MimeType>
            </Format>
            <Format>
              <MimeType>image/geotiff</MimeType>
            </Format>
            <Format>
              <MimeType>application/geotiff</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
    </DataInputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

GRASS and R Integration

GRASS 7.0.svn (nc_spm_08_grass7):~ > R

R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-redhat-linux-gnu (64-bit)

```
> library(spgrass6)
```

```
Loading required package: sp
```

```
Loading required package: XML
```

```
GRASS GIS interface loaded with GRASS version: GRASS 7.0.svn (2013)  
and location: nc_spm_08_grass7
```

```
>
```

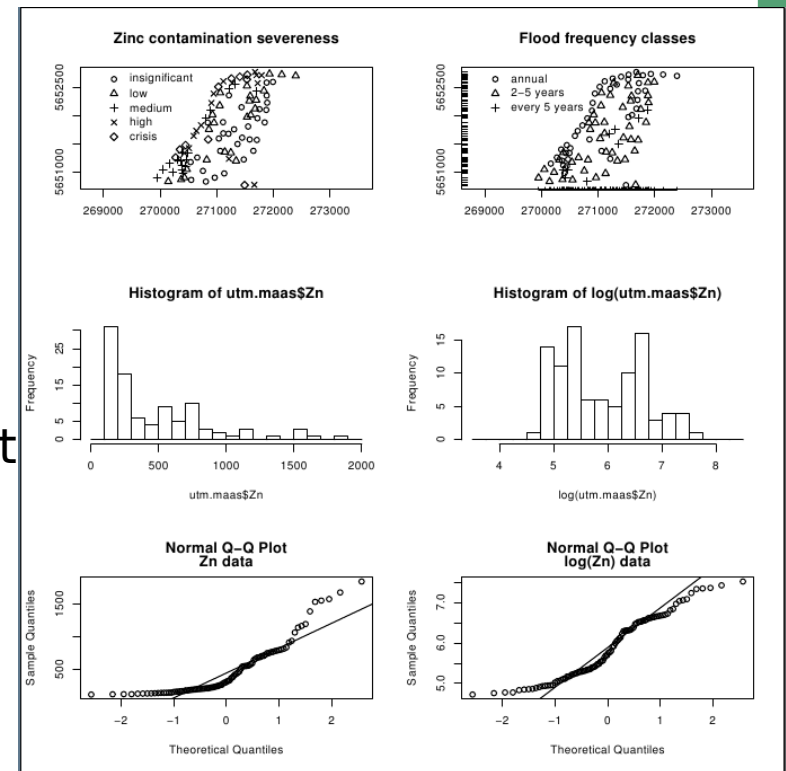
```
> myrast <- readRAST6(c("geology", "elevation"), cat=c(TRUE, FALSE))
```

```
> myvect <- readVECT6("roads")
```

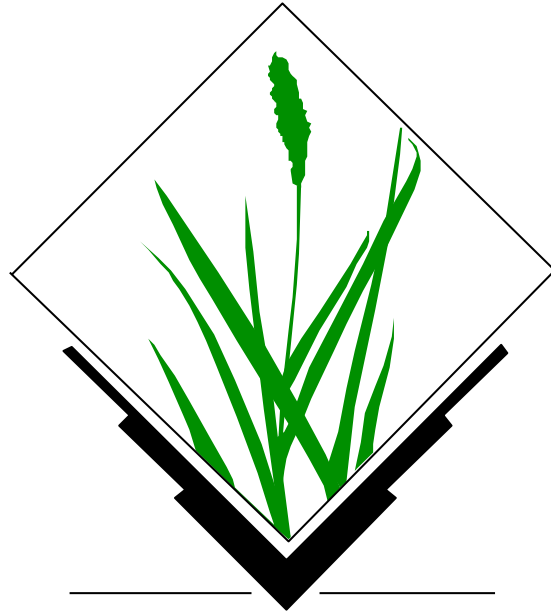
```
...
```

```
> writeRAST6(myrast, "elev_filt", zcol="elev")
```

```
...
```



Visualization



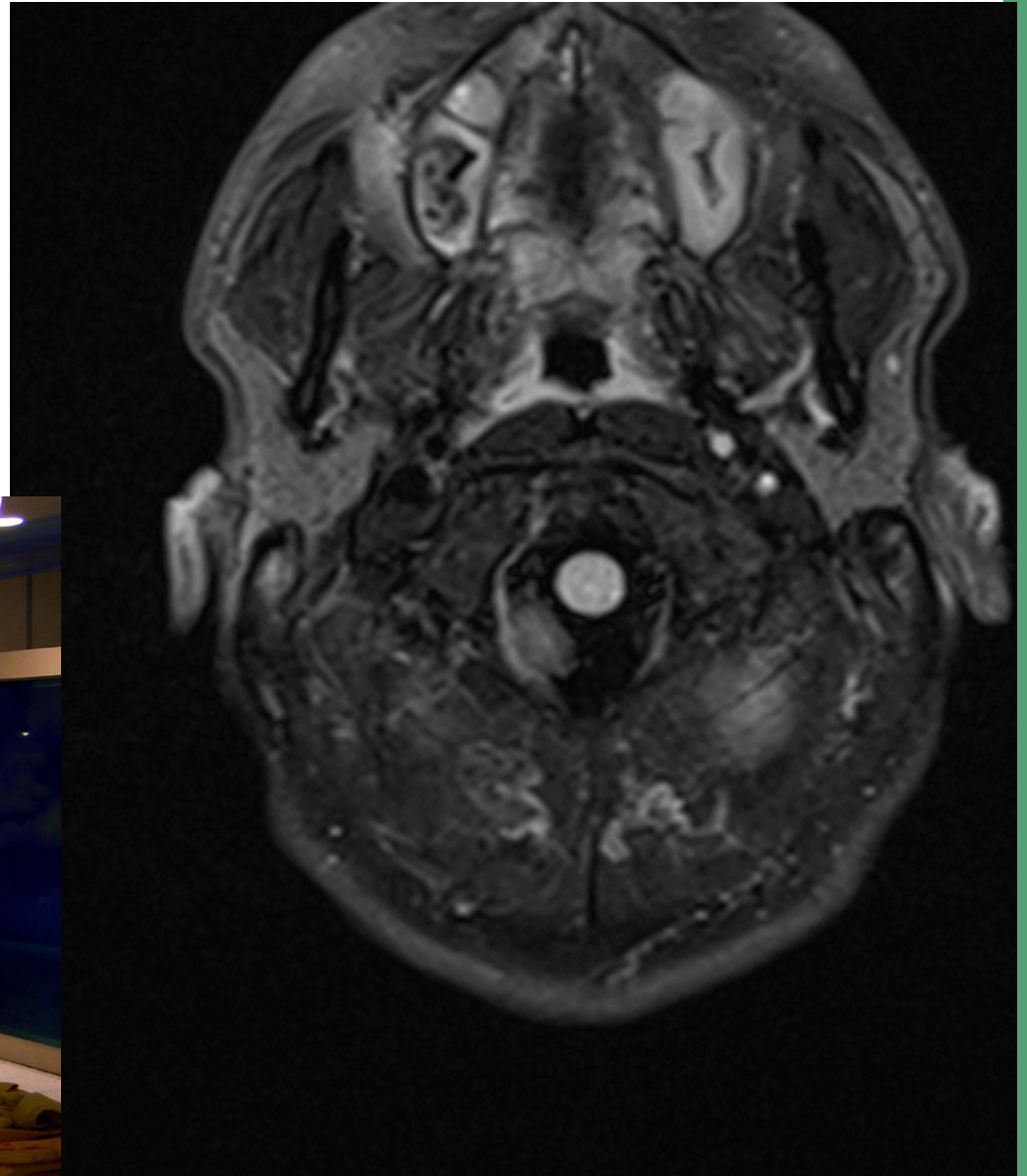
Visualization: GRASS data export to for Augmented Reality



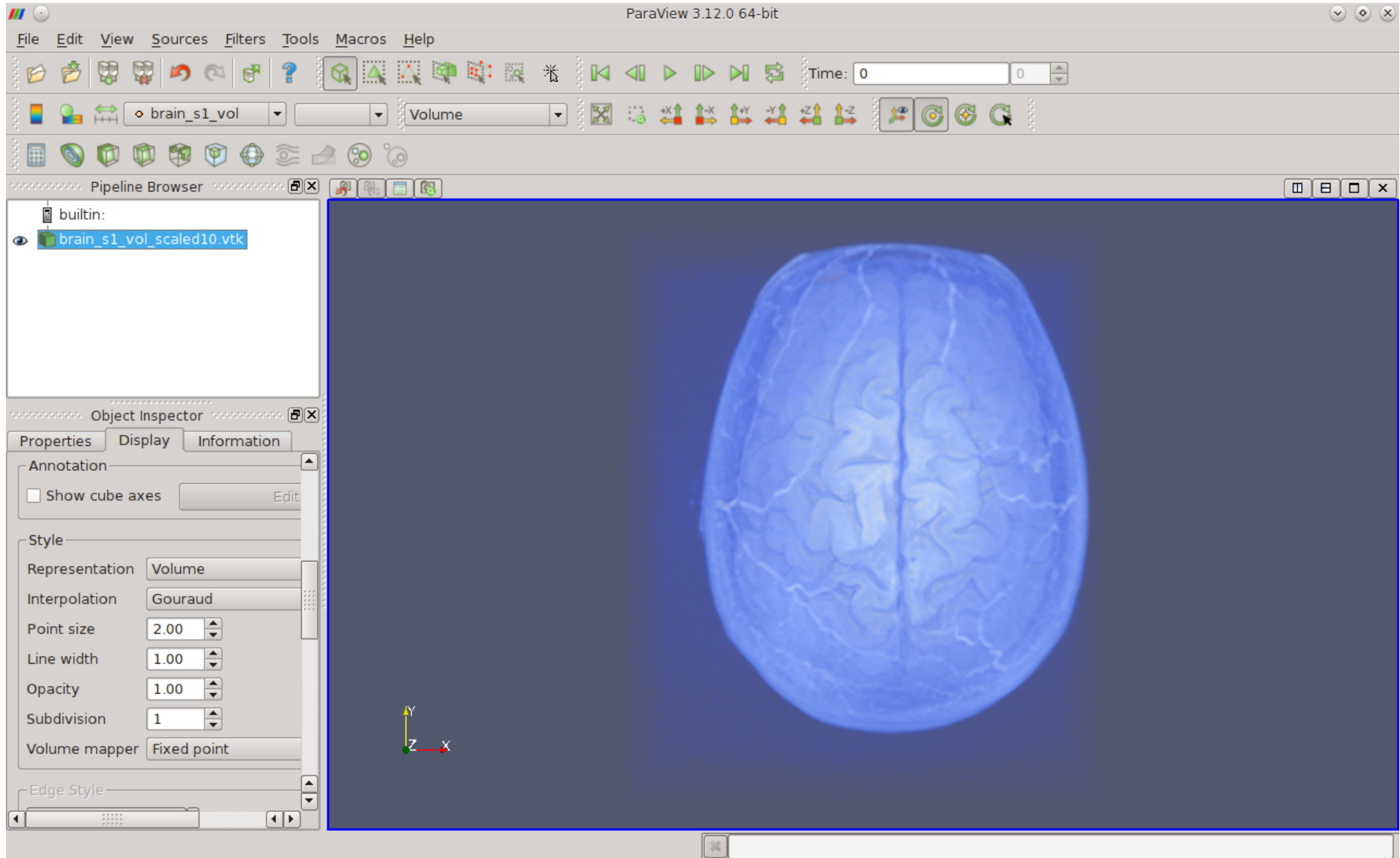
GRASS GIS,
Paraview, and
ARToolkit

A researcher's brain... From MRT scan to Voxels

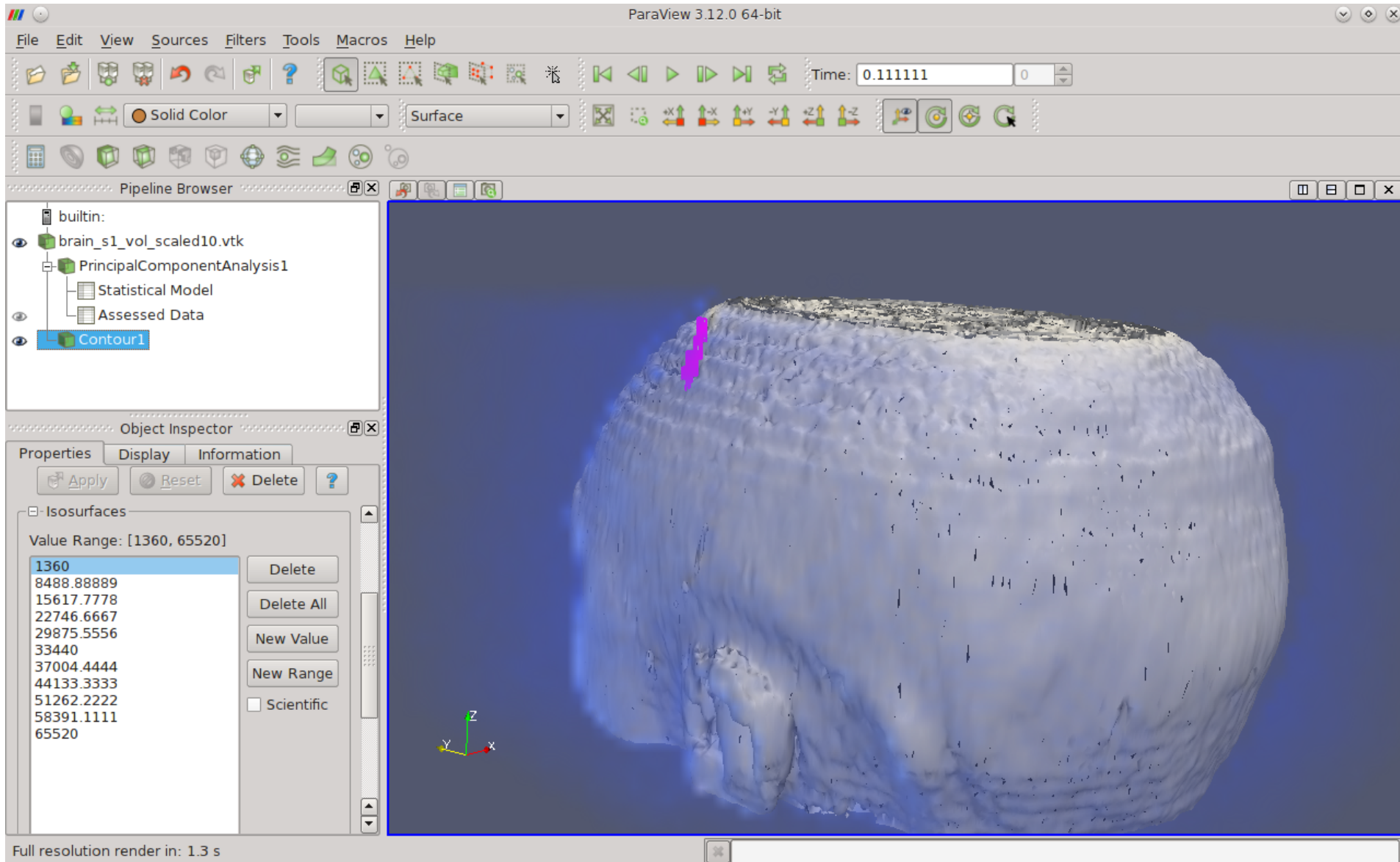
Our FEM colleague
-->



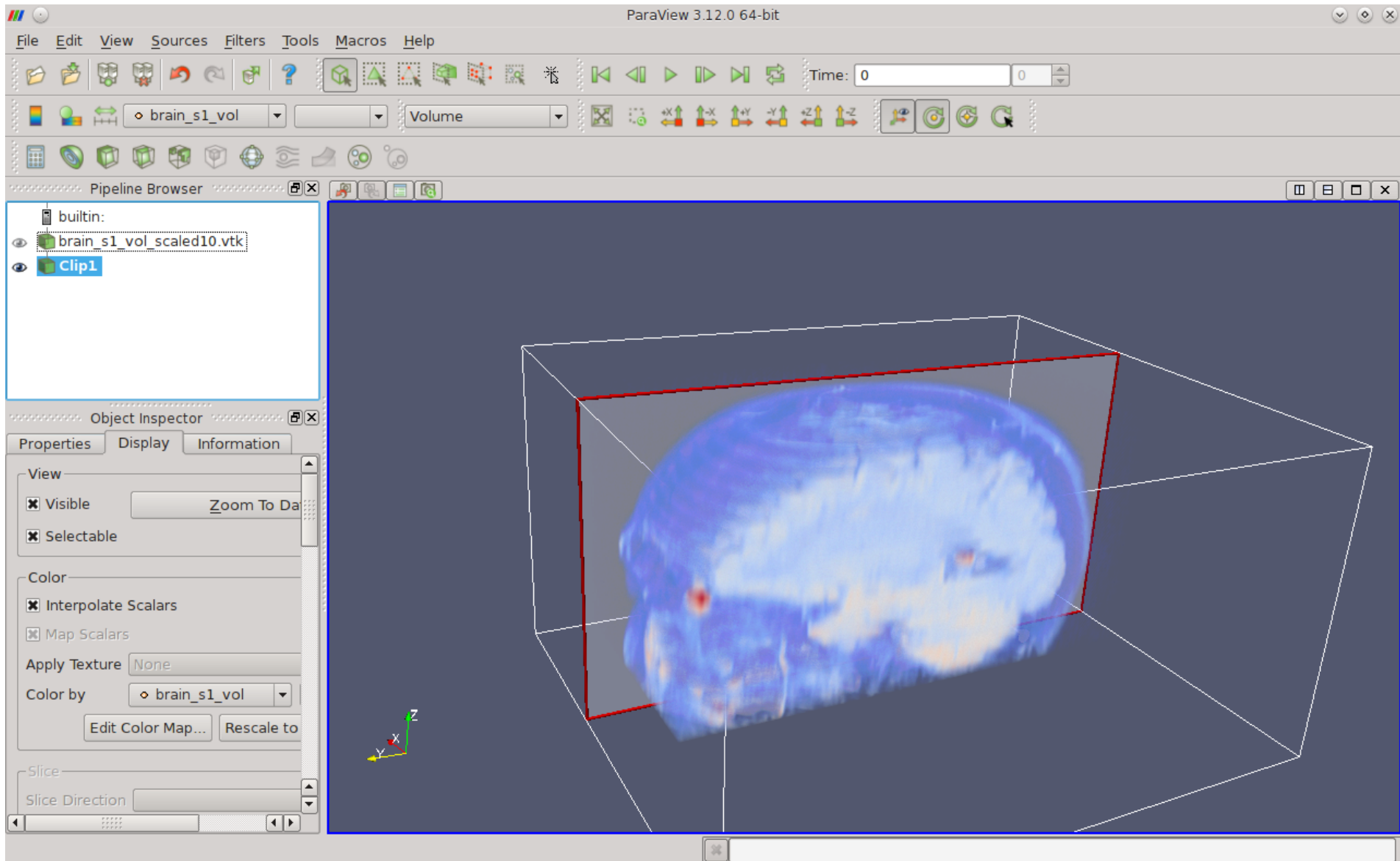
A researcher's brain... From MRT scan to Voxels



A researcher's brain... From MRT scan to Voxels



A researcher's brain... From MRT scan to Voxels



GRASS 7: News in Image processing

Improved modules:

Georectification

Orthorectification

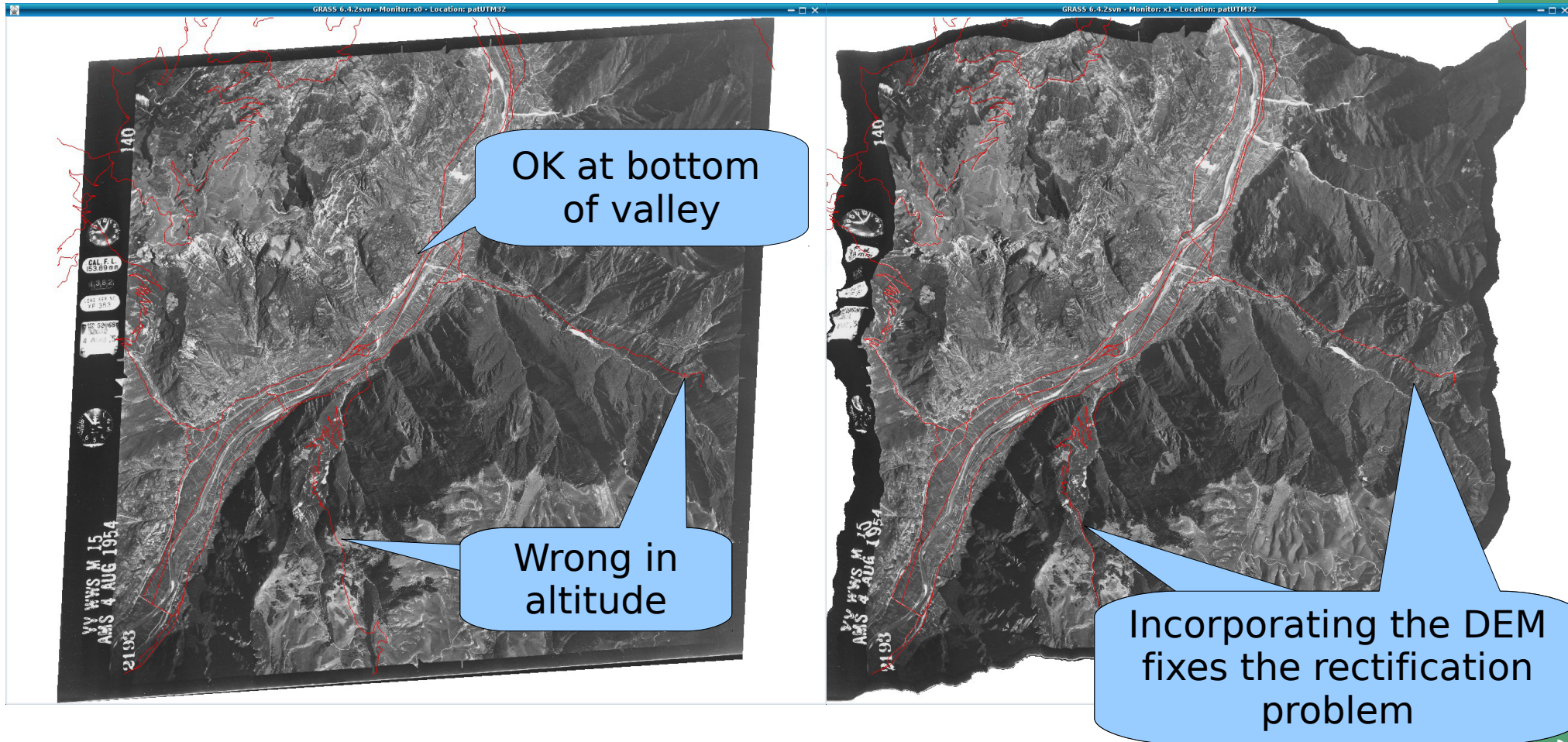
Atmospheric correction

Terrain correction

Landsat cloud detection

Ortho-rectification of aerial images

Why using the DEM for image rectification:

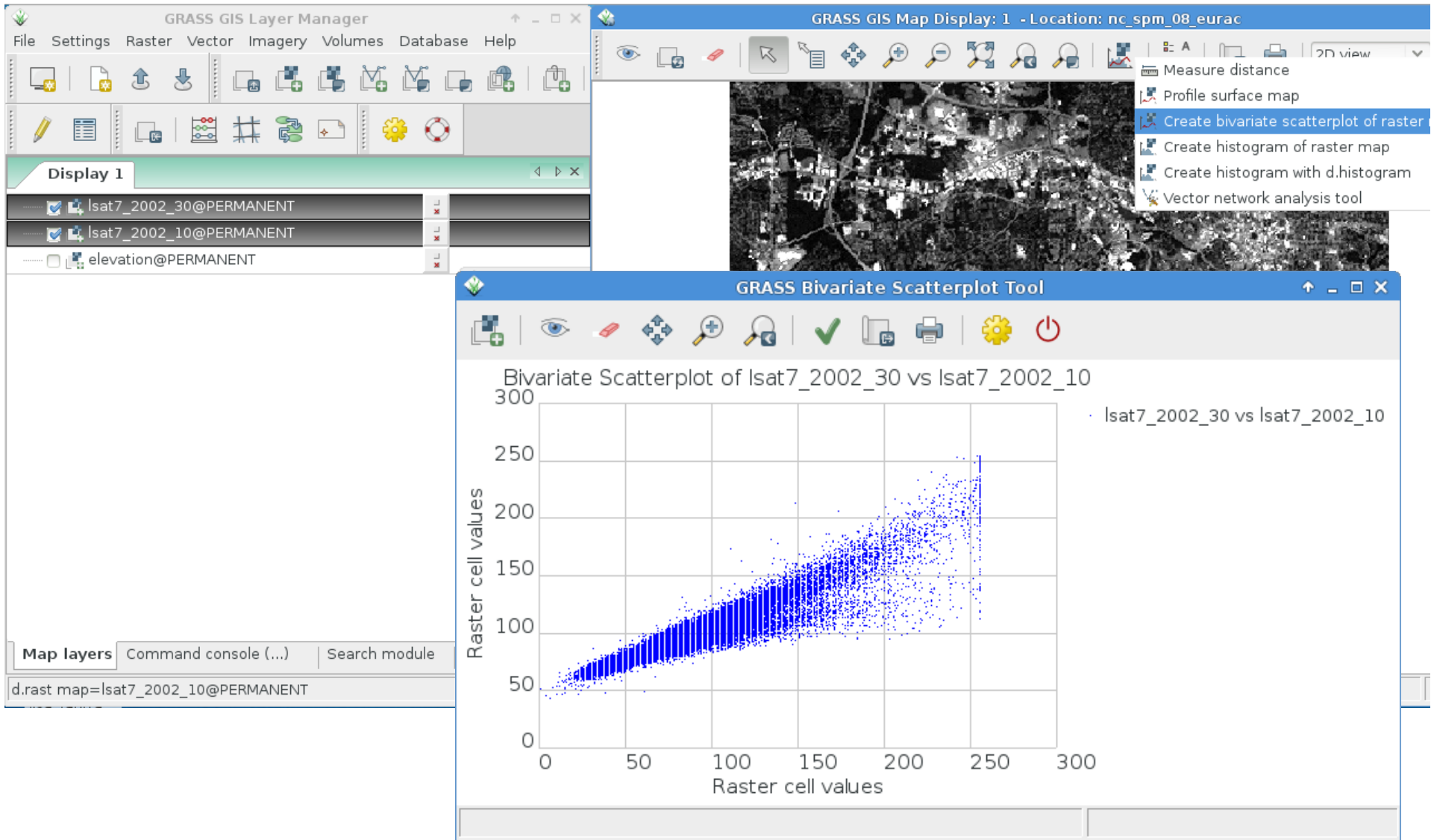


Linear rectification (insufficient)
(Problem commonly seen in Google Earth)

Ortho-rectification

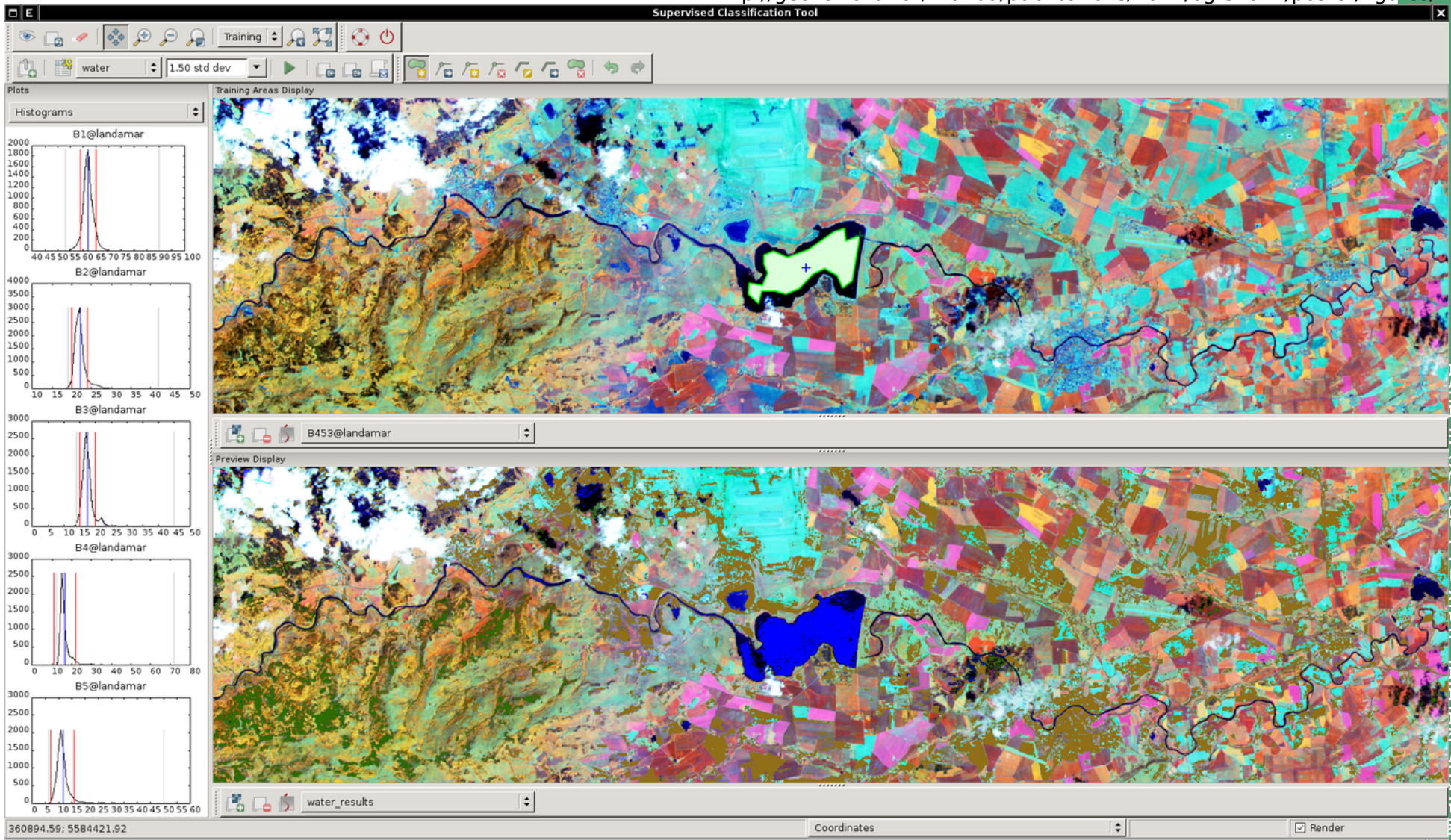
GRASS 7: New bivariate Scatterplots

LANDSAT 7 2002 **channels 1 and 3** of Wake county, NC



Supervised image classification

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

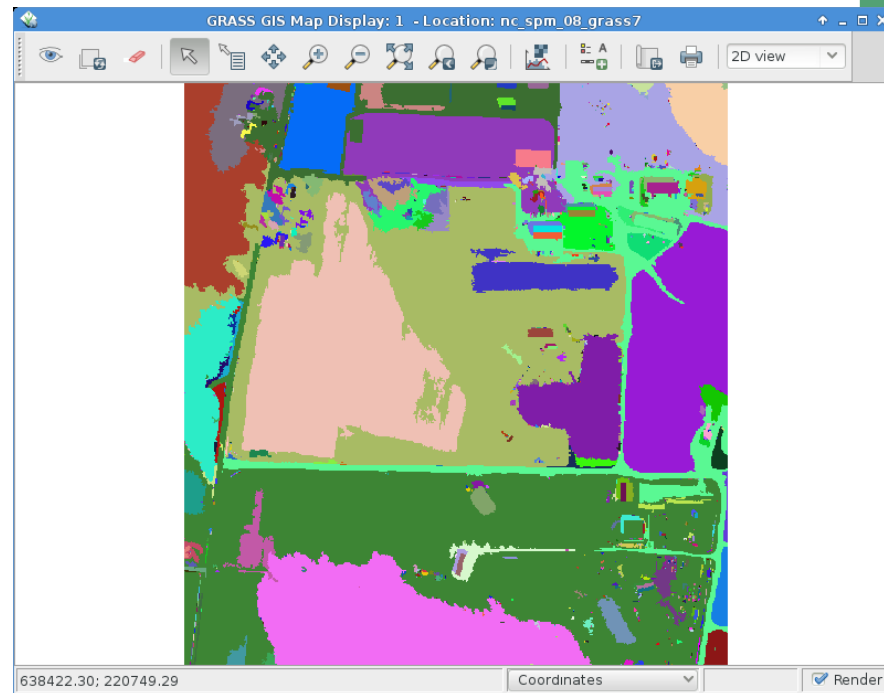
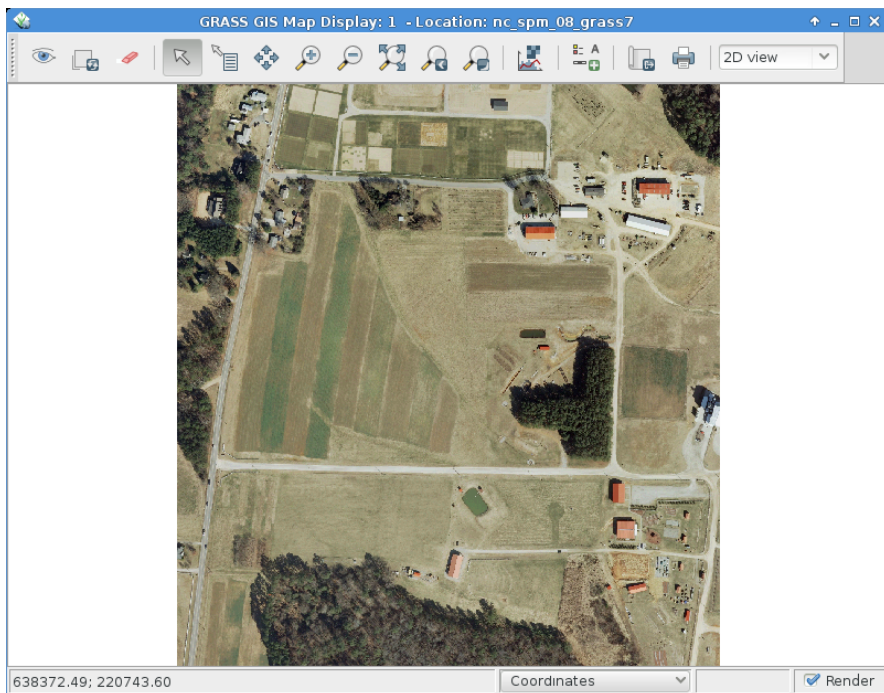


Tool for supervised classification of imagery data.

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

Unsupervised image classification – Segmentation

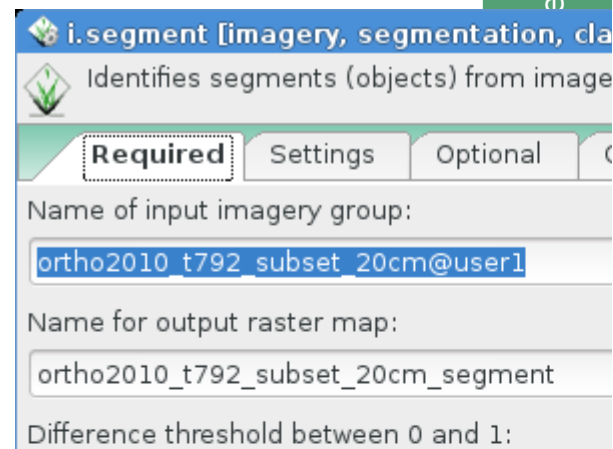
`i.segment` - Identifies segments (objects) from imagery data.



```
# set comp. region to orthophoto but align pixels to elevation map (for speed reasons in  
# this demonstrational exercise – for real work please use the true resolution)  
g.region rast=ortho2010_t792_subset_20cm.blue align=elev_lid792_1m -p
```

```
# Note: the RGB bands are organized as a group  
i.segment group=ortho2010_t792_subset_20cm \  
output=ortho2010_t792_subset_20cm_segment \  
goodness=ortho2010_t792_subset_20cm_seg_fit \  
threshold=0.25
```

Finally vectorize with `r.to.vect`



Where is the stuff?

GRASS GIS Software:

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

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

Addons (user contributed extensions):

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

Free sample data:

Rich data set of North Carolina (NC)

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

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

User Help:

Mailing lists (also in different languages):

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

Wiki:

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

Manuals:

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

<http://grass.osgeo.org>

<http://trac.osgeo.org/grass/wiki/Grass7/NewFeatures>

Markus Neteler

Fondazione E. Mach (FEM)

Centro Ricerca e Innovazione

GIS and Remote Sensing Unit

38010 S. Michele all'Adige (Trento), Italy

<http://gis.cri.fmach.it>

<http://www.osgeo.org>

markus.neteler@fmach.it

neteler@osgeo.org



Enjoy GRASS GIS 7!

THANKS!

