# GRASS GIS 7.4: What's new in a nutshell

Markus Neteler, Veronica Andreo, Luca Delucchi, Martin Landa, Moritz Lennert, Vacla
GRASS Development Team

grass.osgeo.org

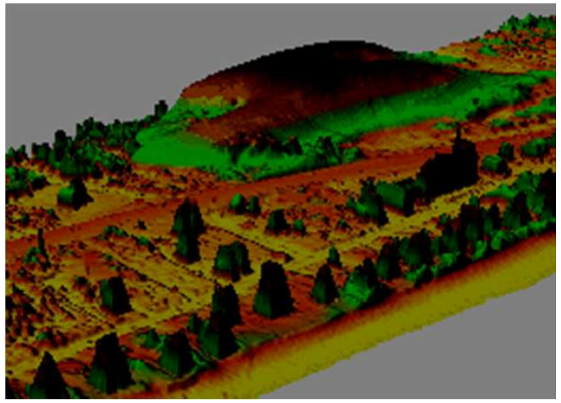
FOSS4G 2018 – Dar es Salaam

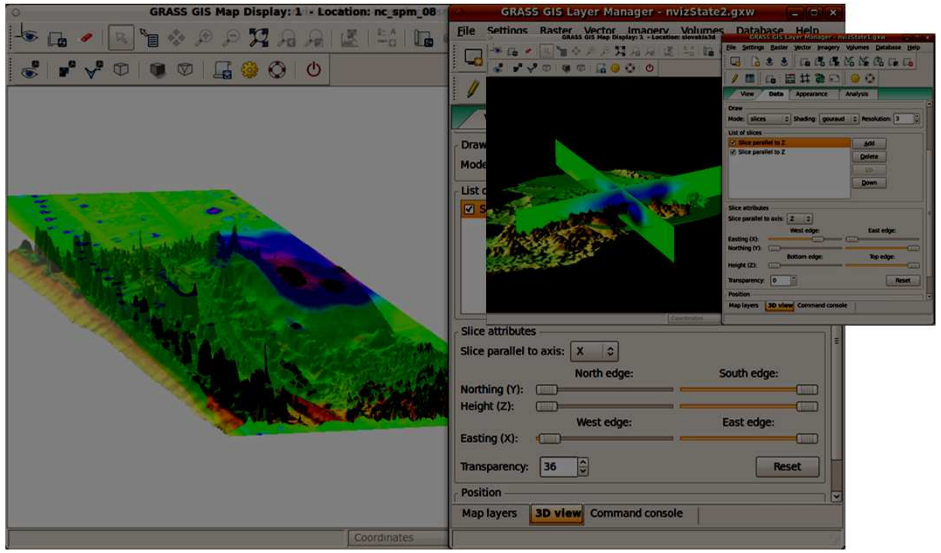
## What is GRASS GIS?

- GRASS GIS is a hybrid, modular GIS software
- GRASS = Geographic Resources Analysis Support System
- GNU General Public License - freely available

- Raster and topological vector data functionality
- 3D raster (voxel) processing
- Image processing
- Visualization options
- Time series analysis

- Portable software ("all" operating systems)
- Graphical user interface and command line


Nags Head LiDAR time series: dune moving o


GRASS GIS Map Display: 1 - Location: nc_spm_08

GRASS GIS Layer Manager - nvizState2.gxw

Slice attributes

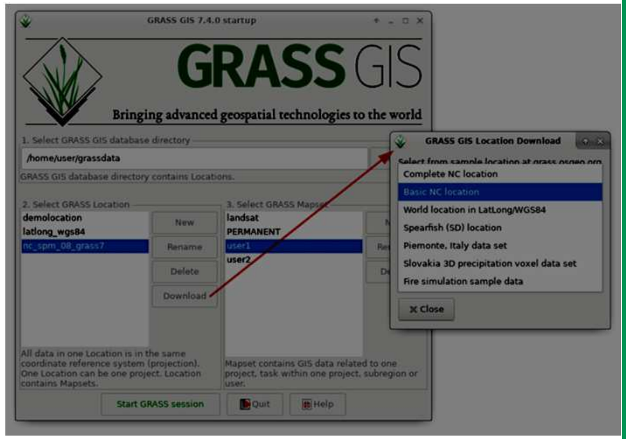
Slice parallel to axis: X

Transparency: 36

## What is new in GRASS GIS 7.4?

New stable version GRASS GIS 7.4

- Improved usability and graphical user interface
- New compression of internal "no data" file which can be huge
- Support for global data extending beyond -180/+180, -90/+90
- Orthorectification with user interface was newly implemented in GRASS GIS 7
- GUI: New *Download* button for sample data
- ... over 480 improvements since G7.2.0


GRASS GIS 7.4.0 startup

GRASS GIS

Bringing advanced geospatial technologies to the world

1. Select GRASS GIS database directory

/home/user/grassdata

GRASS GIS database directory contains Locations.

2. Select GRASS Location

demolocation
latlong_wgs84
nc_spm_08_grass7

All data in one Location is in the same coordinate reference system (projection). One Location can be one project. Location contains Mapsets.

3. Select GRASS Mapset

landsat
PERMANENT
user1
user2

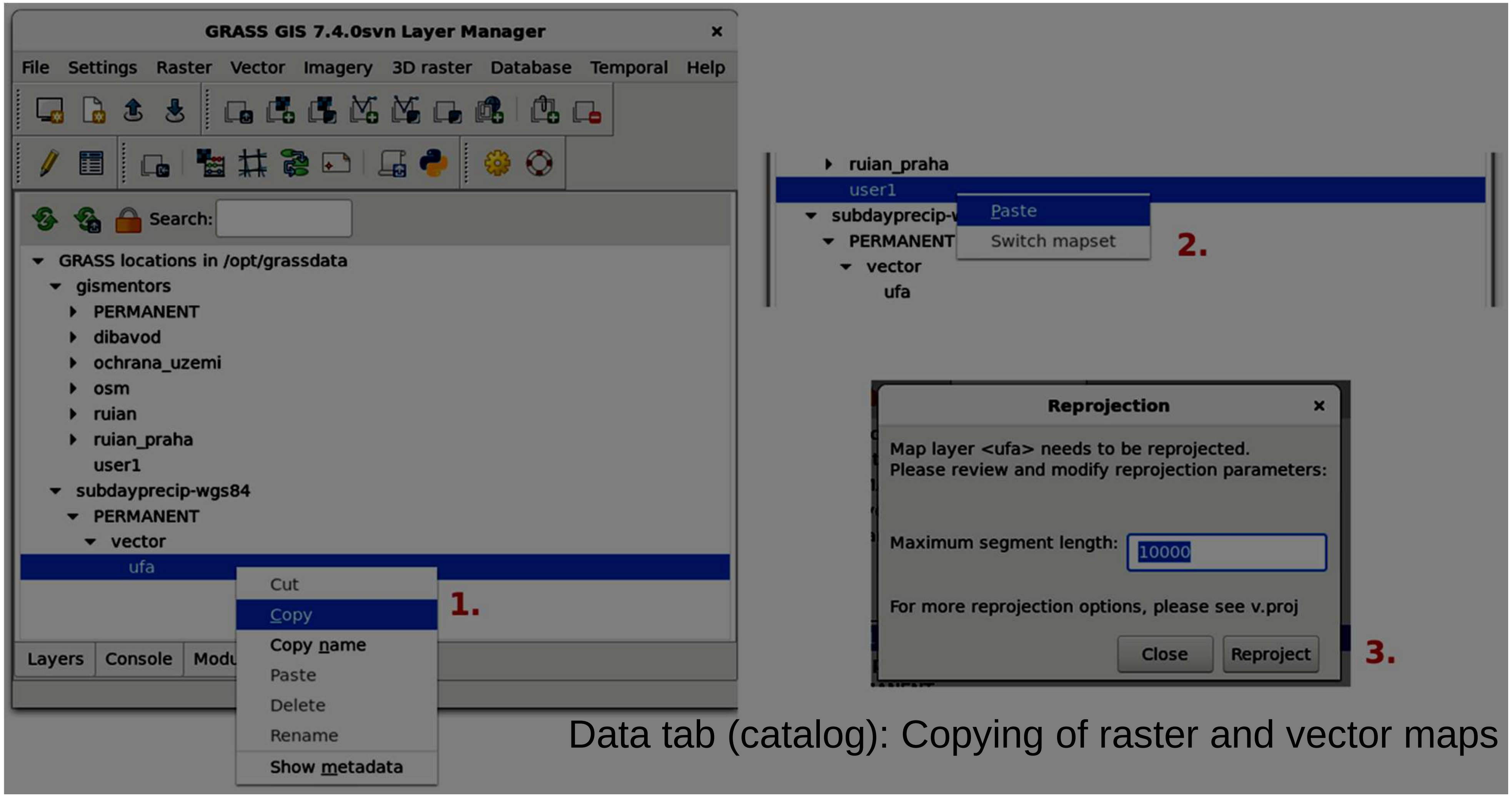
Mapset contains GIS data related to one project, task within one project, subregion or user.

GRASS GIS Location Download

Select from sample location at grass.osgeo.org

Complete NC location
Basic NC location
World location in LatLong/WGS84
Spearfish (SD) location
Piemonte, Italy data set
Slovakia 3D precipitation voxel data set
Fire simulation sample data

# Data catalog improvements


GRASS GIS 7.4.0svn Layer Manager

File Settings Raster Vector Imagery 3D raster Database Temporal Help

Search:

```
▾ GRASS locations in /opt/grassdata
  ▾ gismentors
    ▸ PERMANENT
    ▸ dibavod
    ▸ ochrana_uzemi
    ▸ osm
    ▸ ruian
    ▸ ruian_praha
      user1
  ▾ subdayprecip-wgs84
    ▾ PERMANENT
      ▾ vector
          ufa
```

Cut
Copy
Copy name
Paste
Delete
Rename
Show metadata

1.

Layers Console Modu

```
  ▸ ruian_praha
    user1
▾ subdayprecip-w
  ▾ PERMANENT
    ▾ vector
        ufa
```

Paste
Switch mapset

2.

**Reprojection**

Map layer <ufa> needs to be reprojected.
Please review and modify reprojection parameters:

Maximum segment length: 10000

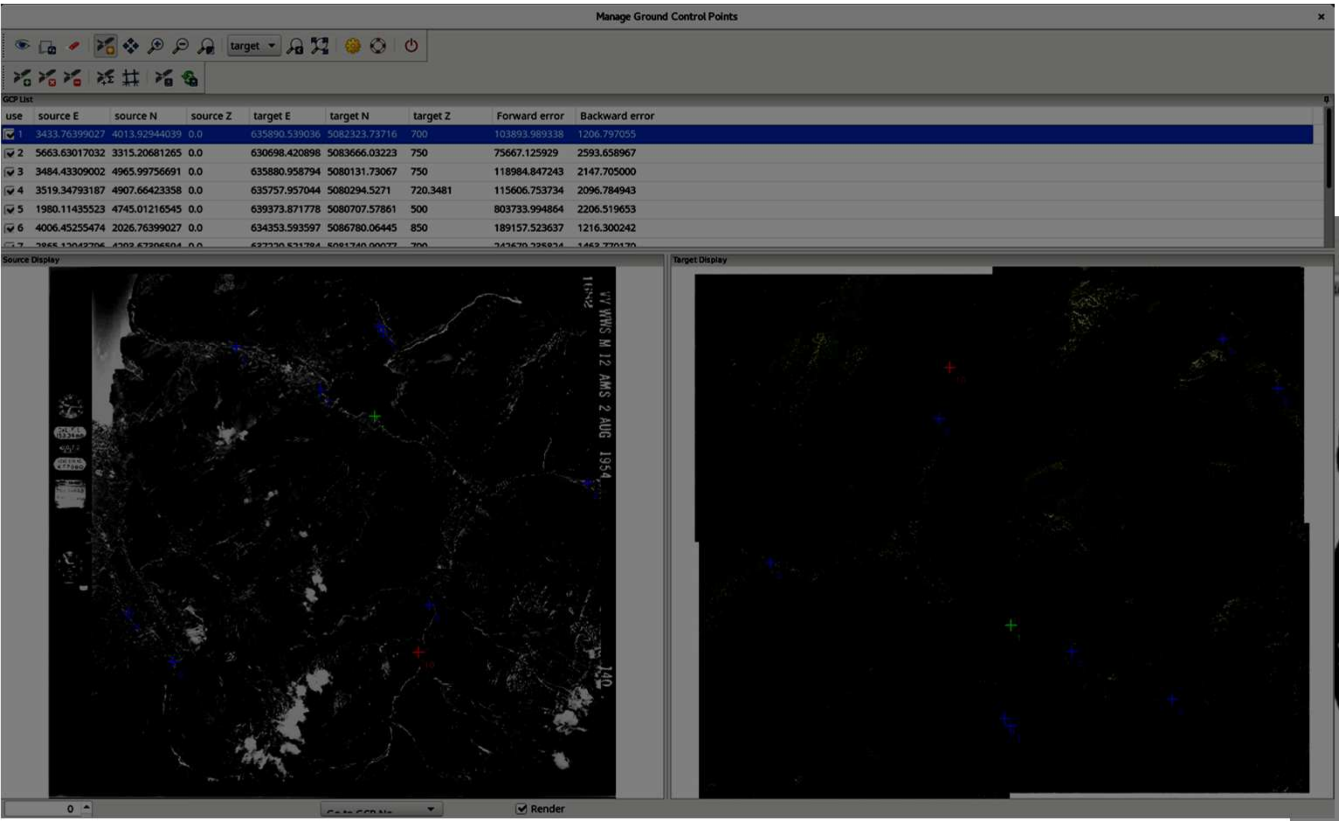
For more reprojection options, please see v.proj

Close | Reproject

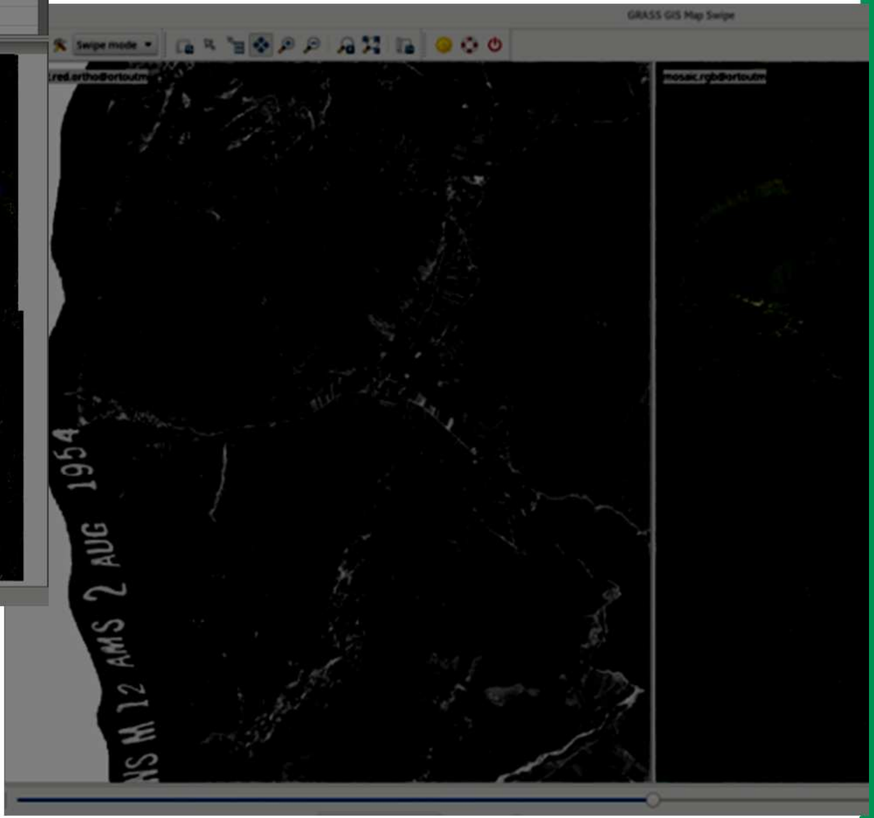
3.

Data tab (catalog): Copying of raster and vector maps betwee

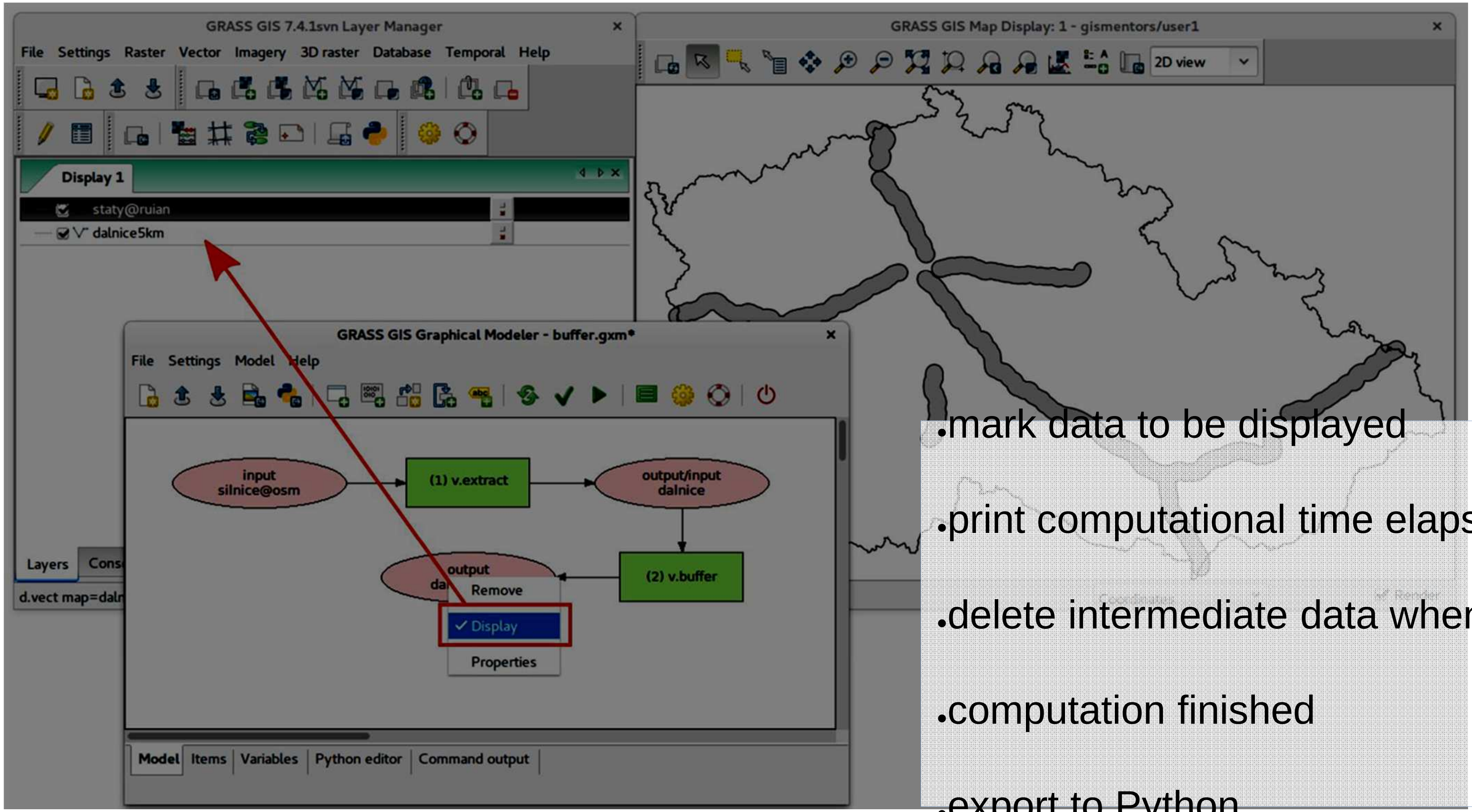
# New Orthorectification GUI



| use | source E | source N | source Z | target E | target N | target Z | Forward error | Backward error |
|---|---|---|---|---|---|---|---|---|
| ✓ 1 | 3433.76399027 | 4013.92944039 | 0.0 | 635890.539036 | 5082323.73716 | 700 | 103893.989338 | 1206.797055 |
| ✓ 2 | 5663.63017032 | 3315.20681265 | 0.0 | 630698.420898 | 5083666.03223 | 750 | 75667.125929 | 2593.658967 |
| ✓ 3 | 3484.43309002 | 4965.99756691 | 0.0 | 635880.958794 | 5080131.73067 | 750 | 118984.847243 | 2147.705000 |
| ✓ 4 | 3519.34793187 | 4907.66423358 | 0.0 | 635757.957044 | 5080294.5271 | 720.3481 | 115606.753734 | 2096.784943 |
| ✓ 5 | 1980.11435523 | 4745.01216545 | 0.0 | 639373.871778 | 5080707.57861 | 500 | 803733.994864 | 2206.519653 |
| ✓ 6 | 4006.45255474 | 2026.76399027 | 0.0 | 634353.593597 | 5086780.06445 | 850 | 189157.523637 | 1216.300242 |
| 7 | 2865.12043706 | 4293.67396504 | 0.0 | 637220.521784 | 5081740.00077 | 700 | 242670.235824 | 1463.770170 |



# Graphical Modeller


GRASS GIS 7.4.1svn Layer Manager
File Settings Raster Vector Imagery 3D raster Database Temporal Help
Display 1
staty@ruian
dalnice5km
Layers
Cons
d.vect map=daln
GRASS GIS Map Display: 1 - gismentors/user1
2D view
GRASS GIS Graphical Modeler - buffer.gxm*
File Settings Model Help
input silnice@osm
(1) v.extract
output/input dalnice
(2) v.buffer
output da
Remove
✓ Display
Properties
Model Items Variables Python editor Command output

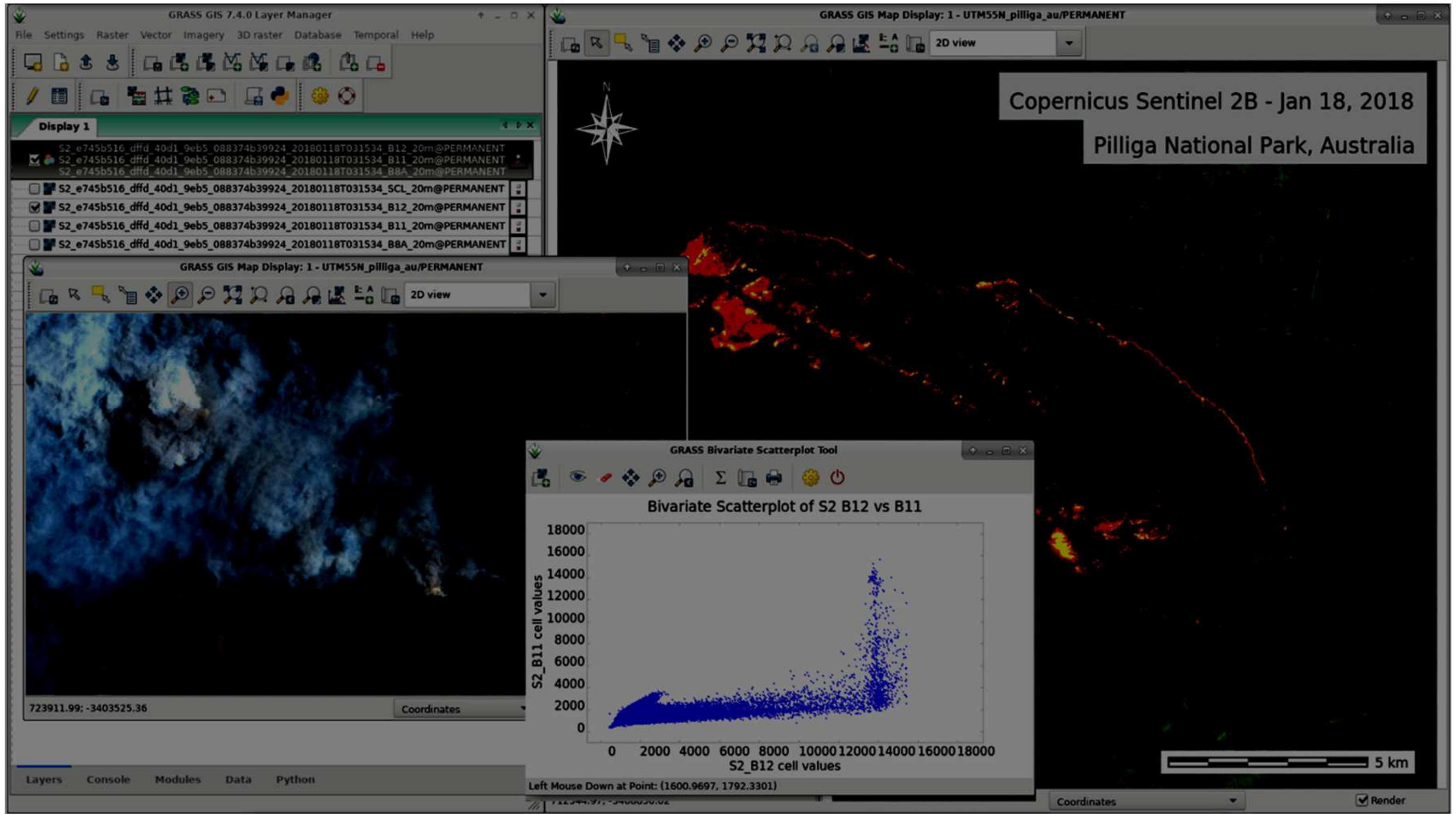
- mark data to be displayed
- print computational time elapsed
- delete intermediate data when
- computation finished
- export to Python

## Copernicus Sentinel-2 processing

New addons: *i.sentinel.download*, *i.sentinel.import*, *i.sentinel.preproc* and *i.se*

Example:

Wildfire in Australia



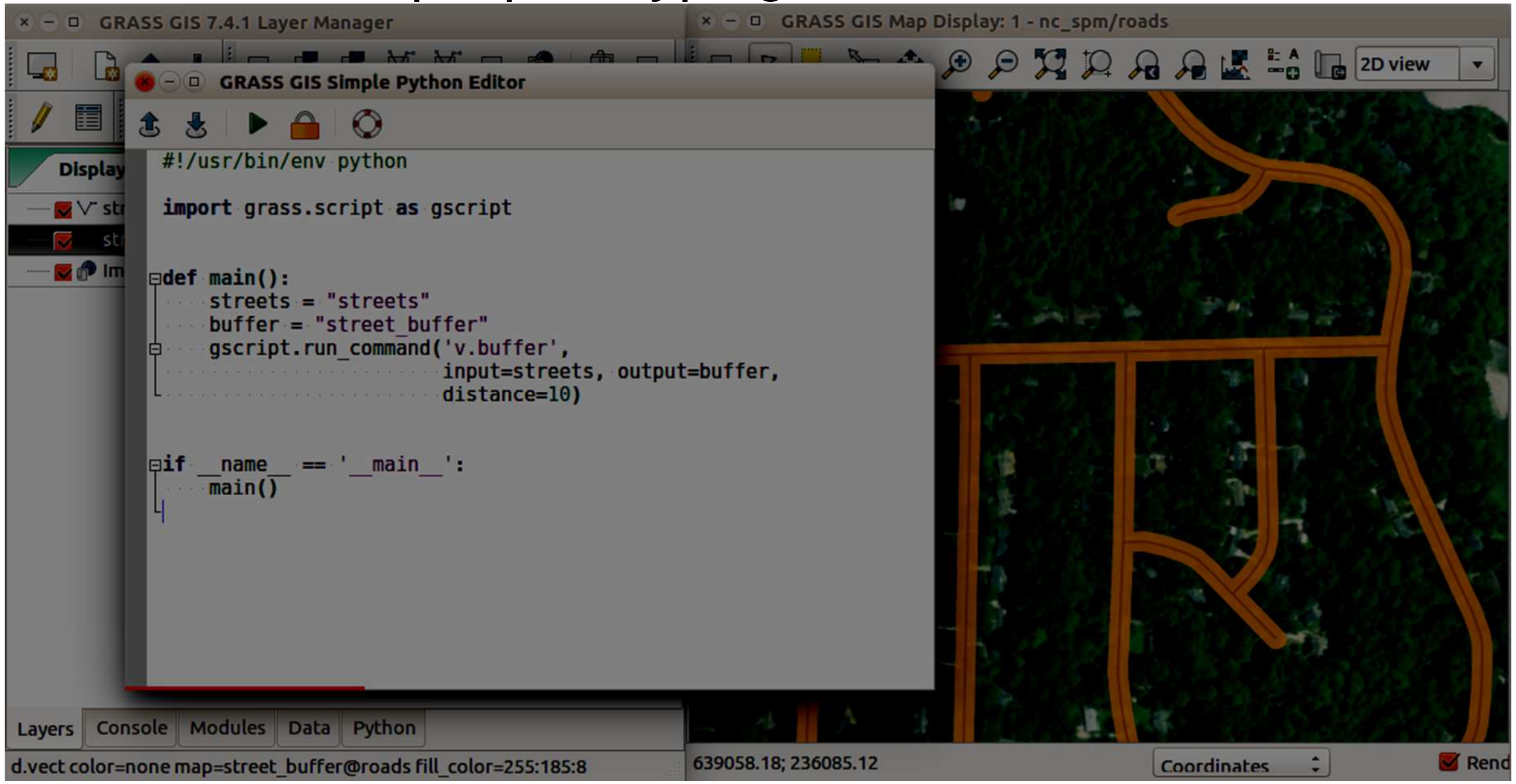
GRASS GIS 7.4.0 Layer Manager
File Settings Raster Vector Imagery 3D raster Database Temporal Help
Display 1
S2_e745b516_dffd_40d1_9eb5_088374b39924_20180118T031534_B12_20m@PERMANENT
S2_e745b516_dffd_40d1_9eb5_088374b39924_20180118T031534_B11_20m@PERMANENT
S2_e745b516_dffd_40d1_9eb5_088374b39924_20180118T031534_B8A_20m@PERMANENT
S2_e745b516_dffd_40d1_9eb5_088374b39924_20180118T031534_SCL_20m@PERMANENT
GRASS GIS Map Display: 1 - UTM55N_pilliga_au/PERMANENT
2D view
Copernicus Sentinel 2B - Jan 18, 2018
Pilliga National Park, Australia
N
5 km
723911.99; -3403525.36
Coordinates
Render
GRASS Bivariate Scatterplot Tool
Bivariate Scatterplot of S2 B12 vs B11
S2_B11 cell values
S2_B12 cell values
0 2000 4000 6000 8000 10000 12000 14000 16000 18000
Left Mouse Down at Point: (1600.9697, 1792.3301)
Layers Console Modules Data Python

# Python Editor

Integrated Python editor for rapid prototyping

Example:

Vector buffer


GRASS GIS 7.4.1 Layer Manager
GRASS GIS Map Display: 1 - nc_spm/roads
GRASS GIS Simple Python Editor
2D view
Display
Layers
Console
Modules
Data
Python
Coordinates
Rend

```
#!/usr/bin/env python

import grass.script as gscript


def main():
    streets = "streets"
    buffer = "street_buffer"
    gscript.run_command('v.buffer',
                        input=streets, output=buffer,
                        distance=10)


if __name__ == '__main__':
    main()
```

```
d.vect color=none map=street_buffer@roads fill_color=255:185:8
```

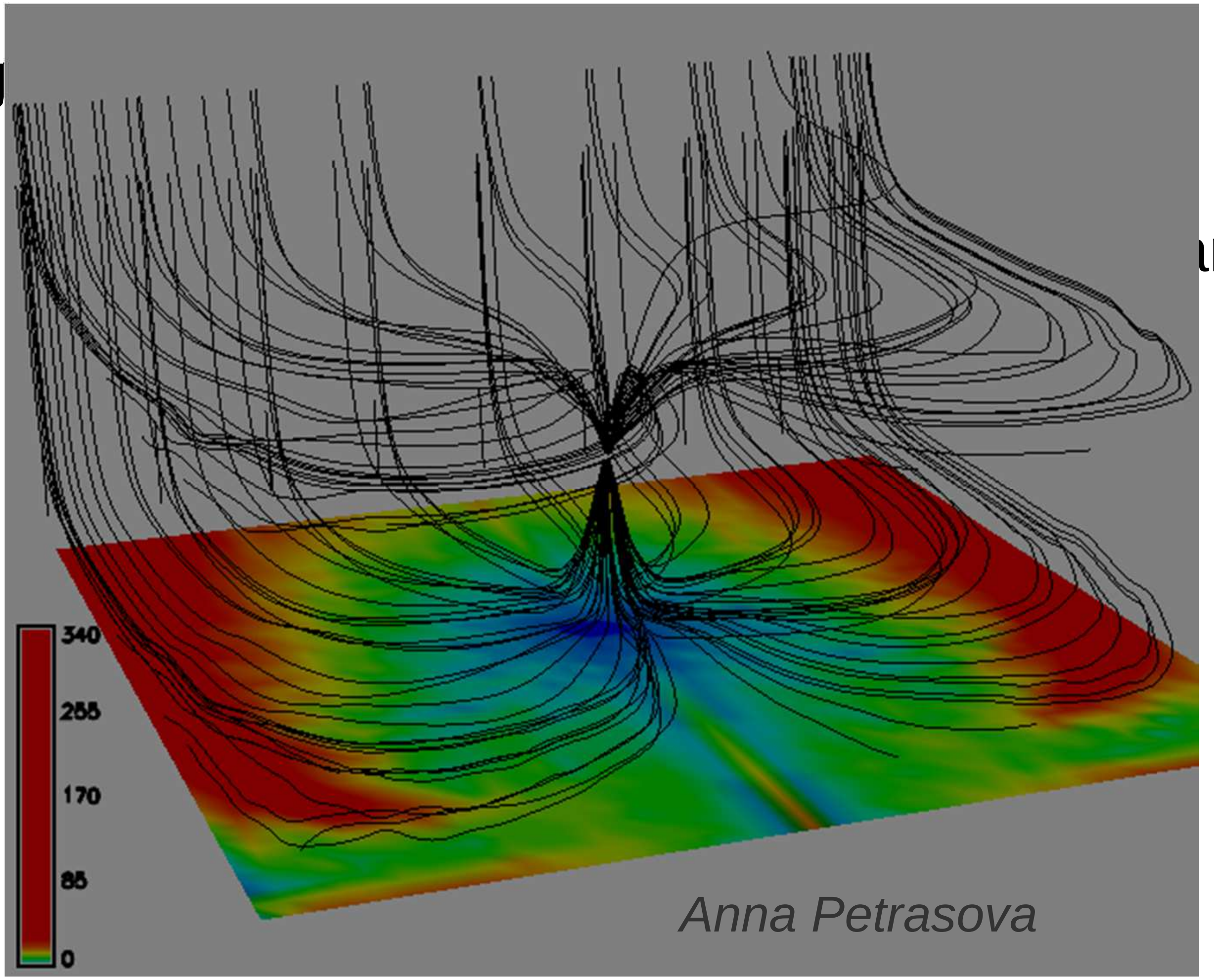
639058.18; 236085.12

*Vaclav Petras*

## 3D raster gradients and flowlines

3D raster (voxel) processing

*r3.flow* and *r3.gradient* to co                                                                                        nd re



## TGRASS: *t.rast.algebra* and *t.rast3d.algebra*: temporal algebra

Compute annual hydro-thermal coefficients (HTC) from daily climate data

$$HTC = \frac{\sum P_{(T>10^\circ C)}}{\sum T_{(T>10^\circ C)} \cdot \frac{1}{10}}$$

T := daily temperatures,
P := daily precipitation

~ **60 years of daily data**, each pixel in time = virtual meteo station

```
t.rast.algebra "HTC = (D {+,contains,1} if(T >= 10, P, 0)) /
                      (D {+,contains,1} if(T >= 10, T / 10, 0))"
```

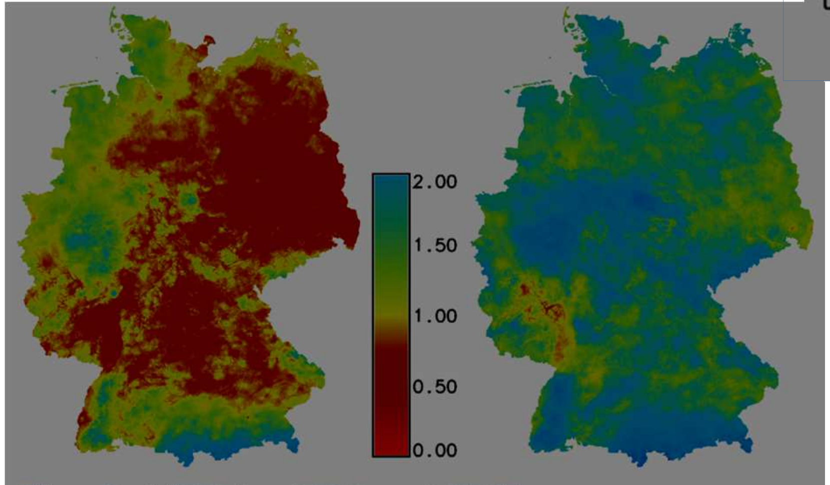


**Fig. 6:** HTC for 2003 and 2007

*Leppelt & Gebbert, EGU 2015*

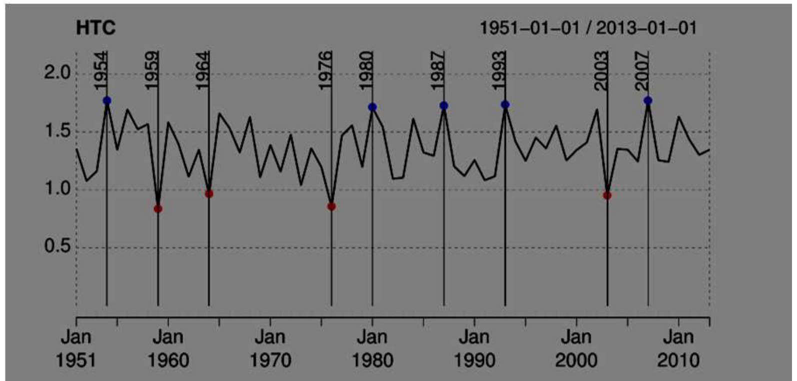


**Fig. 7:** HTC of extreme events for droughts (HTC $< 1$) in red and humid years (HTC $> 1.7$ ) in blue

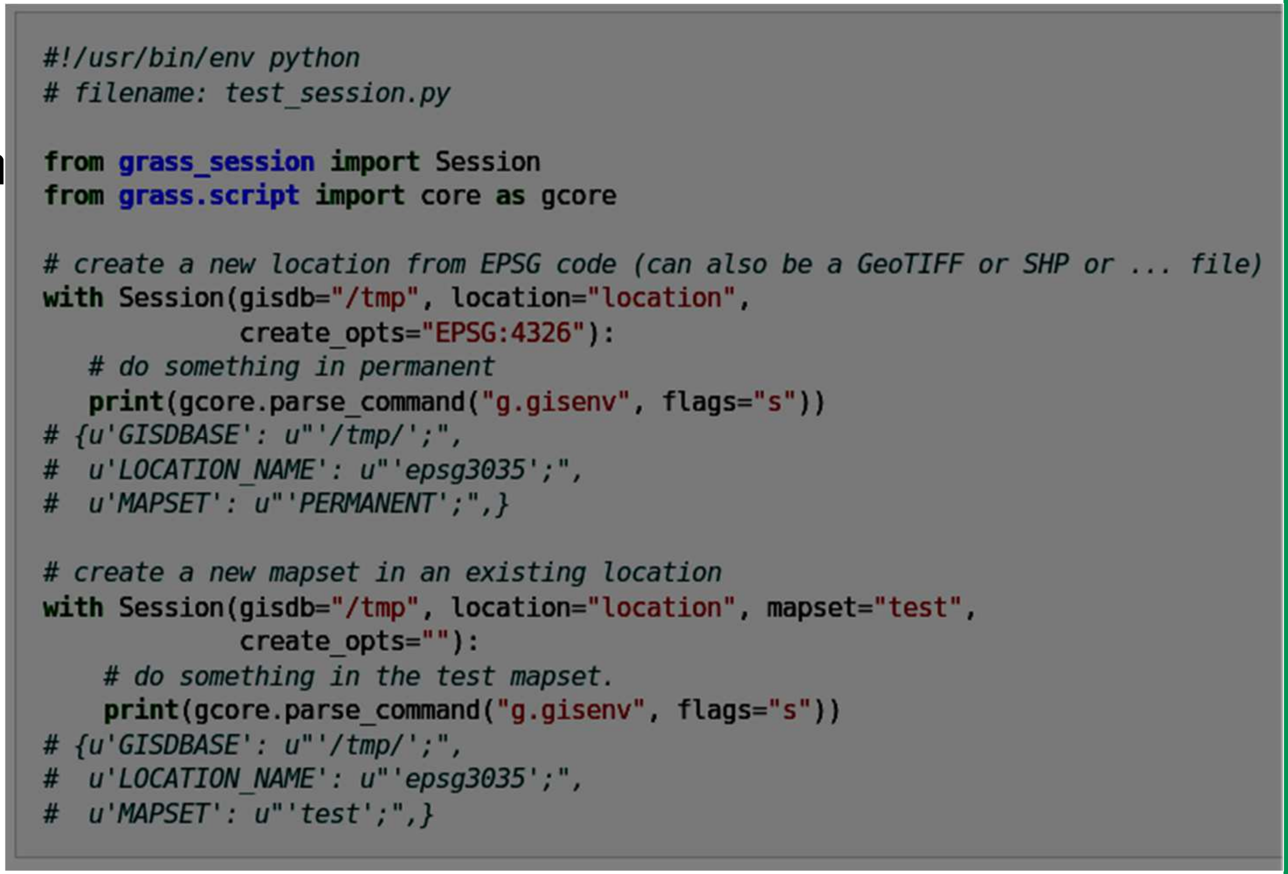
## GRASS GIS and Python

Using GRASS GIS from “outside” through “grass-session”

```
pip install grass-session
```

Now it’s easy to use GRASS GIS
as a processing backend in Python!

Combine with GDAL, OTB, ...



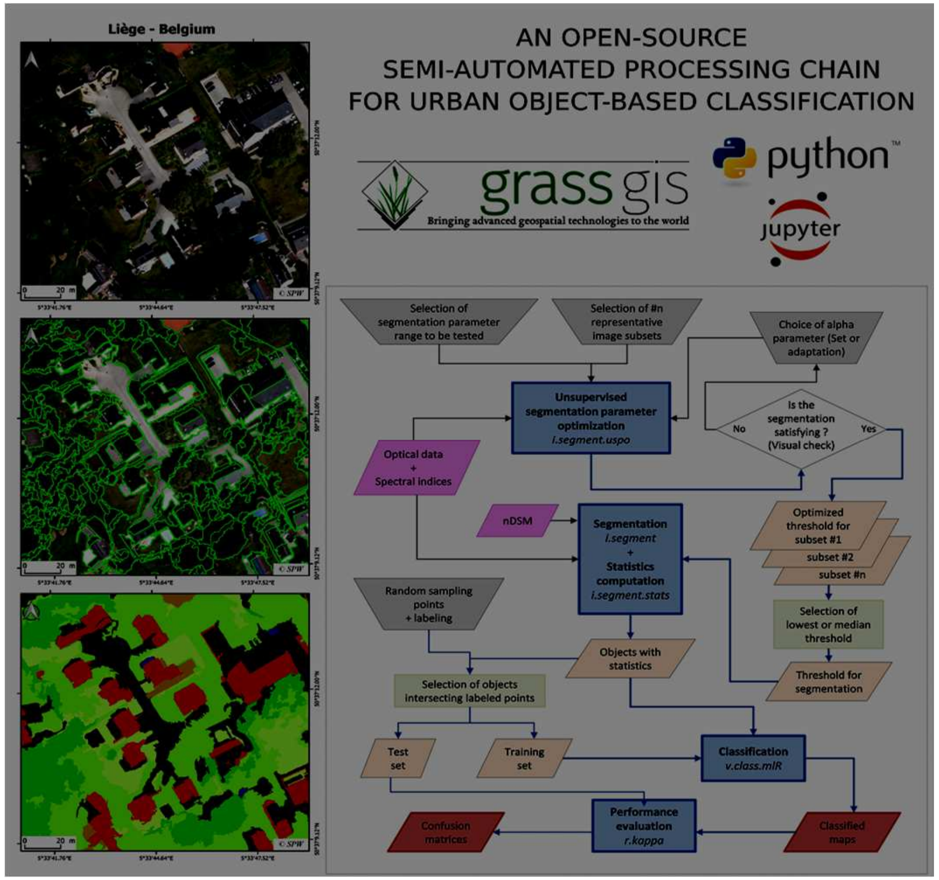
```
#!/usr/bin/env python
# filename: test_session.py

from grass_session import Session
from grass.script import core as gcore

# create a new location from EPSG code (can also be a GeoTIFF or SHP or ... file)
with Session(gisdb="/tmp", location="location",
             create_opts="EPSG:4326"):
   # do something in permanent
   print(gcore.parse_command("g.gisenv", flags="s"))
# {u'GISDBASE': u"'/tmp/';",
#  u'LOCATION_NAME': u"'epsg3035';",
#  u'MAPSET': u"'PERMANENT';",}

# create a new mapset in an existing location
with Session(gisdb="/tmp", location="location", mapset="test",
             create_opts=""):
    # do something in the test mapset.
    print(gcore.parse_command("g.gisenv", flags="s"))
# {u'GISDBASE': u"'/tmp/';",
#  u'LOCATION_NAME': u"'epsg3035';",
#  u'MAPSET': u"'test';",}
```

# Remote sensing in GRASS GIS: object-based image analysis



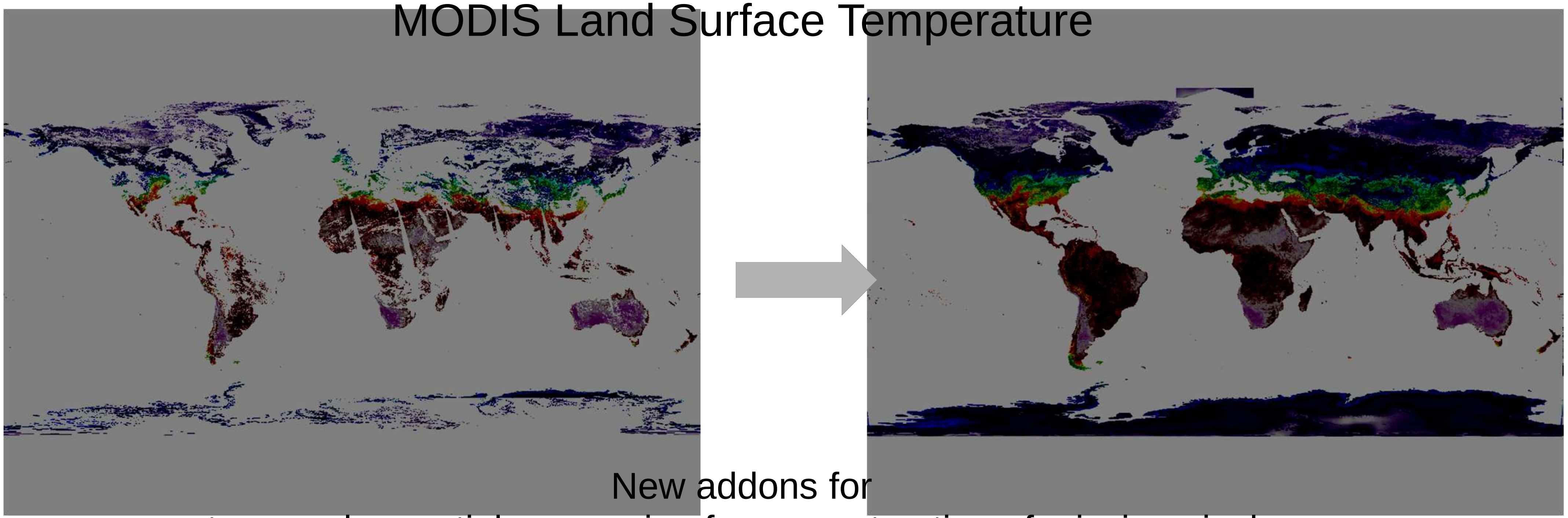
Liège - Belgium
AN OPEN-SOURCE SEMI-AUTOMATED PROCESSING CHAIN FOR URBAN OBJECT-BASED CLASSIFICATION
grass gis
Bringing advanced geospatial technologies to the world
python
jupyter
Selection of segmentation parameter range to be tested
Selection of #n representative image subsets
Choice of alpha parameter (Set or adaptation)
Unsupervised segmentation parameter optimization i.segment.uspo
Is the segmentation satisfying ? (Visual check)
No
Yes
Optical data + Spectral indices
nDSM
Segmentation i.segment + Statistics computation i.segment.stats
Optimized threshold for subset #1
subset #2
subset #n
Selection of lowest or median threshold
Threshold for segmentation
Random sampling points + labeling
Objects with statistics
Selection of objects intersecting labeled points
Test set
Training set
Classification v.class.mlR
Performance evaluation r.kappa
Confusion matrices
Classified maps

Source : http://dx.doi.org/10.3390/rs9040358

- Complete toolchain from segmentatio
- Including
  - unsupervised segmentation paramete
  - high performance object statistics cal
  - module-level parallelization
- Recently created module for SLIC sup

## High-performance computing

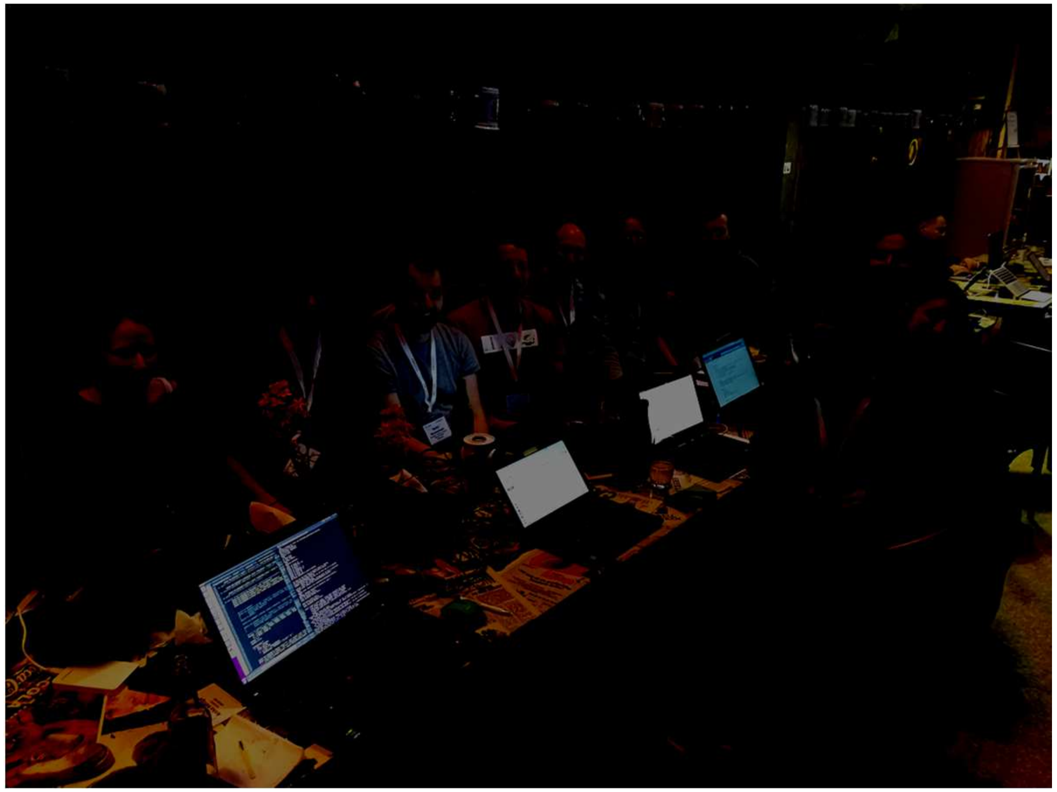
MODIS Land Surface Temperature



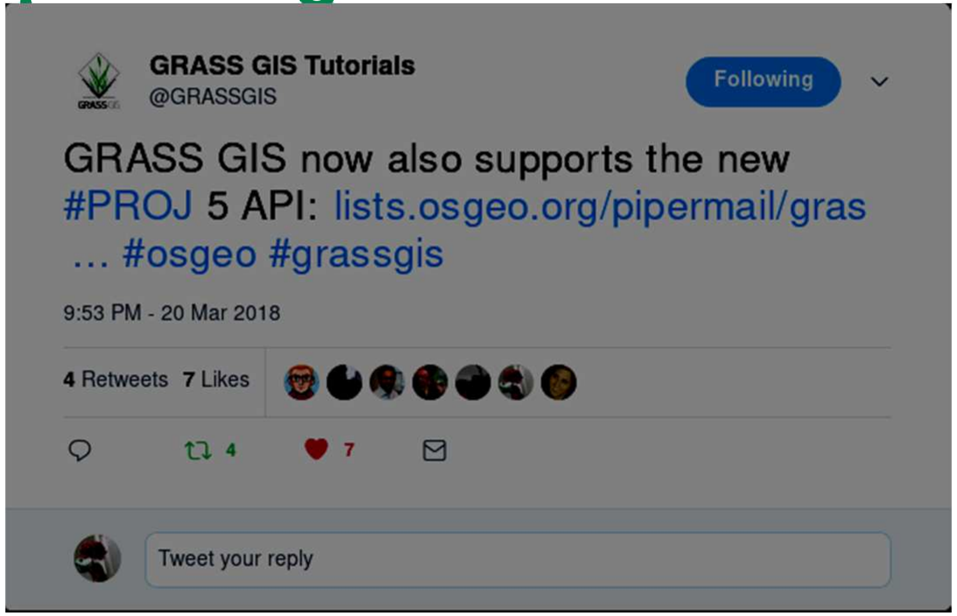
New addons for
temporal + spatial processing for reconstruction of missing pixels

Data: https://zenodo.org/record/1135230

## Community activities: Code Sprint 2018 at FOSSGIS Bonn Basecamp – Integration



20 March 2018



GRASS GIS Tutorials
@GRASSGIS

GRASS GIS now also supports the new #PROJ 5 API: lists.osgeo.org/pipermail/gras … #osgeo #grassgis

9:53 PM - 20 Mar 2018

4 Retweets 7 Likes

## Community activities: Google Code-IN for 13-17 year old pre-university students

https://grasswiki.osgeo.org/wiki/GRASS_GCI_Ideas_2017

3.1 **Install** GRASS GIS on your computer and download North Carolina dataset

3.2 **Compile** GRASS GIS

3.3 Add examples and/or screenshots to different **manual** pages

3.4 Add **test suites** to different modules

3.5 **Designs**

3.5.1 Splash screen for GRASS GIS GUI start-up

3.5.2 T-shirt for 2018 Code Sprint

3.5.3 Banner for location wizard

3.6 **Blog** entry about GRASS GIS

3.7 **Videos**

3.7.1 How to create a location

*Around 120 students*

## Community activities: GSoC 2018

Google Summer of Code 2018

https://trac.osgeo.org/grass/wiki/GSoC/2018

OSS-Fuzz - Continuous **Fuzzing** for Open Source Software for GRASS GIS
Implement a series of **image fusion** algorithms in GRASS GIS
Enhance 3D **rendering** capabilities in GRASS GIS
Additional functionality for running GRASS GIS modules in **Jupyter** Notebook
Integration of **PDAL** into GRASS GIS
**Benchmarking** framework for GRASS GIS
GRASS GIS as a post-processing part of **WebODM**
Additional **GUI** tools for image analysis
Module to create quadtree **tiling**
Tools for generating **unit tests** from examples in the manual
**Mapnik** rendering engine for GRASS GIS
Generalized GUI code for **Qt-based GUI**
GRASS GIS **3D viewer** NVIZ module independent of the main GUI
Integration of v.profile into **GUI** profiling tool
Add **CMake** build system for GRASS GIS
Add a cloud masking module for **Sentinel** data in GRASS GIS
Full support of **Python 3** in GRASS GIS
Improve GRASS integration in **QGIS 3**
New easy-to-use CLI and **API** for GRASS GIS

## Thanks for your attention!

GRASS GIS

grass.osgeo.org