Helena Mitasova
Maria Brovelli
Roberto Antolín
Markus Neteler

Power user workshop

GRASS image processing of optical and Lidar data

FOSS4G2006 Conference
11-15 Sept. 2006, Lausanne, Switzerland

OSGeo.org
The Open Source Geospatial Foundation

GRASS Image processing

- Part I: Optical image processing (Markus Neteler)
- Part II: Lidar processing (Helena Mitasova)
- Part III: Lidar processing (Maria Brovelli and Roberto Antolín

How to use the GFOSS liveCD?

KNOPPIX is a Live-Linux CDROM/DVD which contains all relevant software and an excellent hardware-detection algorithm – even network and sound card detection should work.

Note: If needed, insert USB sticks before starting the PC.
KNOPPIX does not modify your hard disk.
It only runs from the RAM memory.

The workshop CD was prepared by Stephan Holl and contains GFOSS software and the sample data sets.

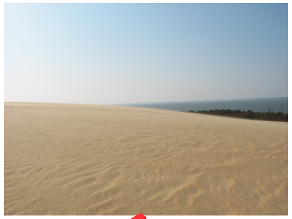
How to use the GFOSS liveCD?

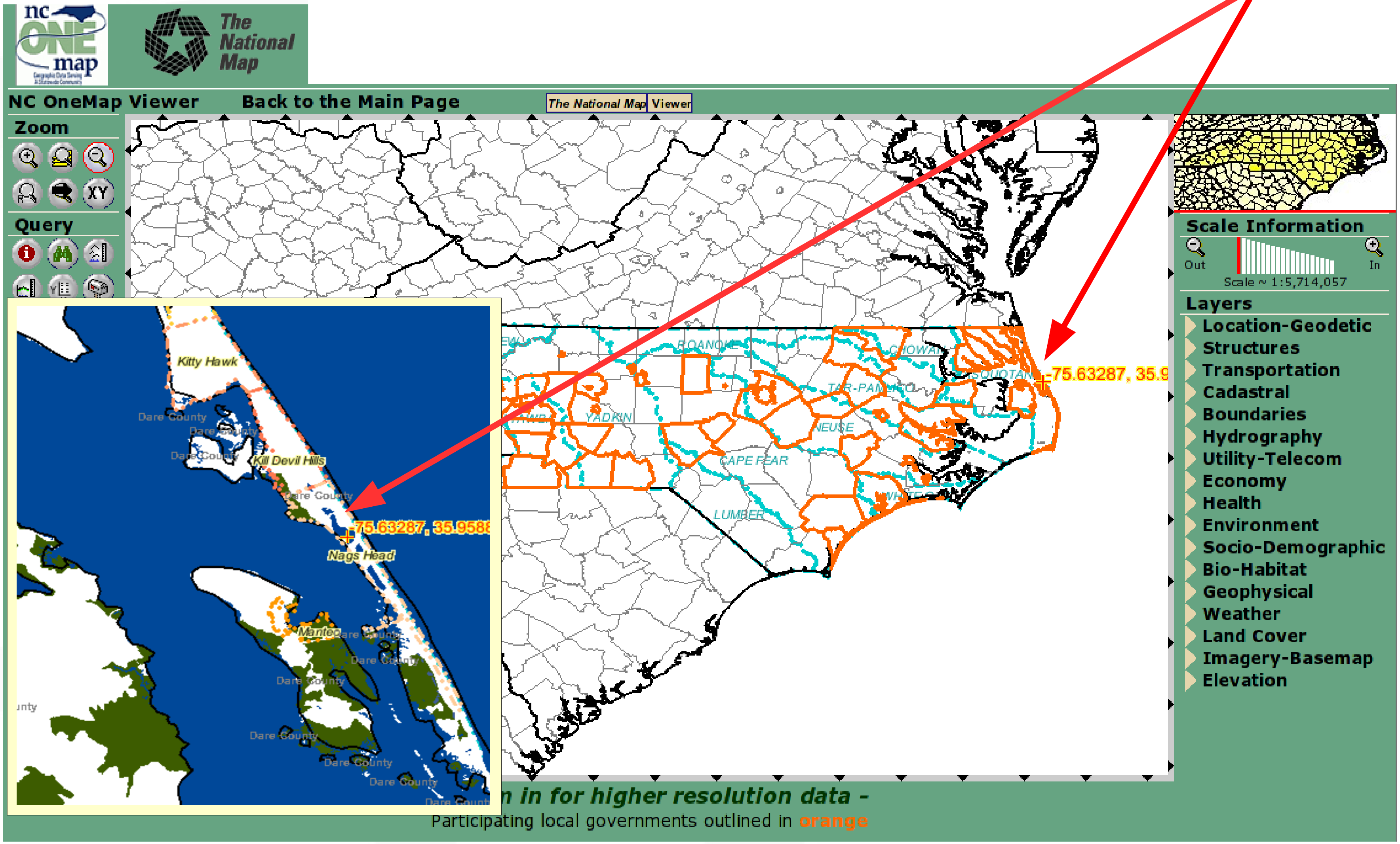
To use the liveCD is as easy as this:

- (Re)boot your PC with the CDROM. This will start Linux without installation, just from CDROM. (Check BIOS settings to enable boot-from-CD)
- At “boot” of the GISIX CDROM, enter <return> (here it is also possible to select another language than English):

  boot: **`<enter>`**
- After little time you arrive at the graphical desktop.
- Take a look at the menu (“K” at lower left in the main menu bar)

North Carolina Sample data set: Jockey's Ridge





North Carolina - Jockey's Ridge

Photo: H. Mitasova

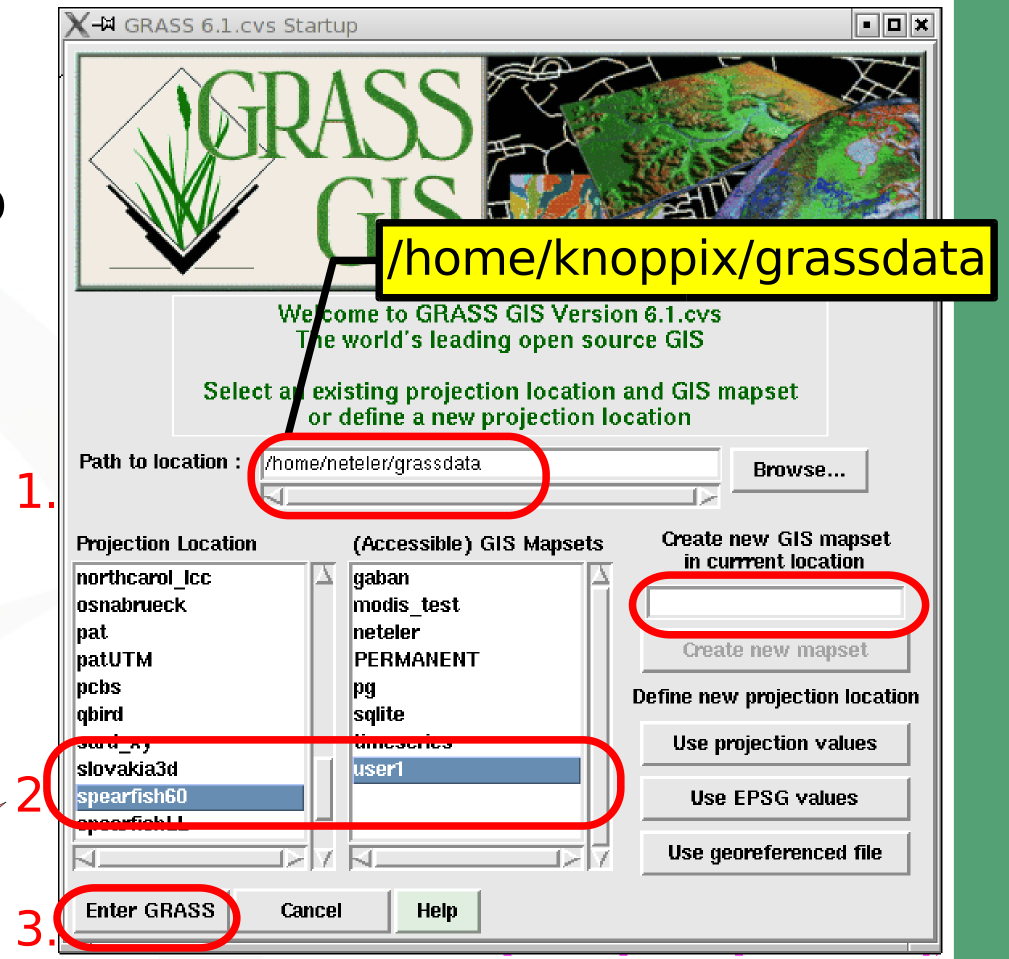
Starting GRASS

- Start a "terminal" (shell) to enter commands

- Start GRASS 6 within the terminal window:

  # get startup help:
  grass6 - help

  # start with GUI:
  grass6 -gui



Location:
nc_utm18
Mapset:
create your own

Download, reproject and import

1. LANDSAT7 data set was downloaded from GLCF Maryland (GeoTIFF data are in UTM projection, size around 650MB)

   ftp://ftp.glcf.umiacs.umd.edu/glcf/Landsat/WRS2/p014/r035/p014r035_7x19990923.ETM-EarthSat-Orthorectified/

2. During the workshop, we use a small spatial subset which was cut out with the GDAL tool “gdal_translate”.

3. Expand data set “tarball” (package containing all data):

```
cd /tmp/

# enter in 1 line:
tar xvfz $HOME/workshop_material/GRASS-imagery/
  landsat7_NC_p014r035_1999_subset.tar.gz
```

Import the LANDSAT7 Scene

Bulk import of all channels into the NC UTM18N sample location:

```
for i in *.img ; do
  NAME=`echo $i | cut -d'_' -f4`
  r.in.gdal $i out=$NAME
done
```

This imports the subsequent channels:

- Vis: B, G, R (# 10, 20, 30) – 28.5m
- NIR (# 50) – 28.5m
- MIR (# 70) – 28.5m
- TIR (# 61, 62) – thermal, low/high gain – 57m
- PAN (# 80) – panchromatic – 14.25m

Visualization of color composites

RGB display

To start, set the region to the blue channel (res. is 28.5m):

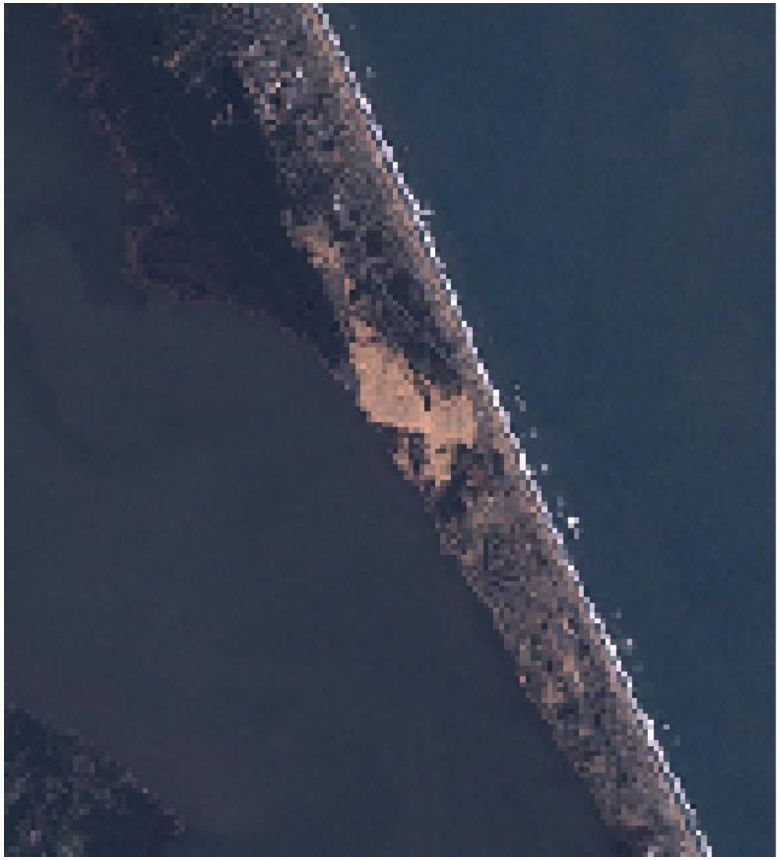
```
g.region rast=nn10 -p
```

Open a GRASS monitor:

```
d.mon x0
```

Generate the color composite:

```
d.rgb b=nn10 g=nn20 r=nn30
```



Visualization of color composites

Color enhancement: strip away outliers

Note: the color enhancement works better for larger regions

```
i.landsat.rgb b=nn10 g=nn20 r=nn30
d.rgb b=nn10 g=nn20 r=nn30
```

To revert, do for all channels:

```
r.colors <map> col=grey
```

Save composite to new map

```
r.composite b=nn10 \
   g=nn20 r=nn30 \
   out=landsat.rgb
```

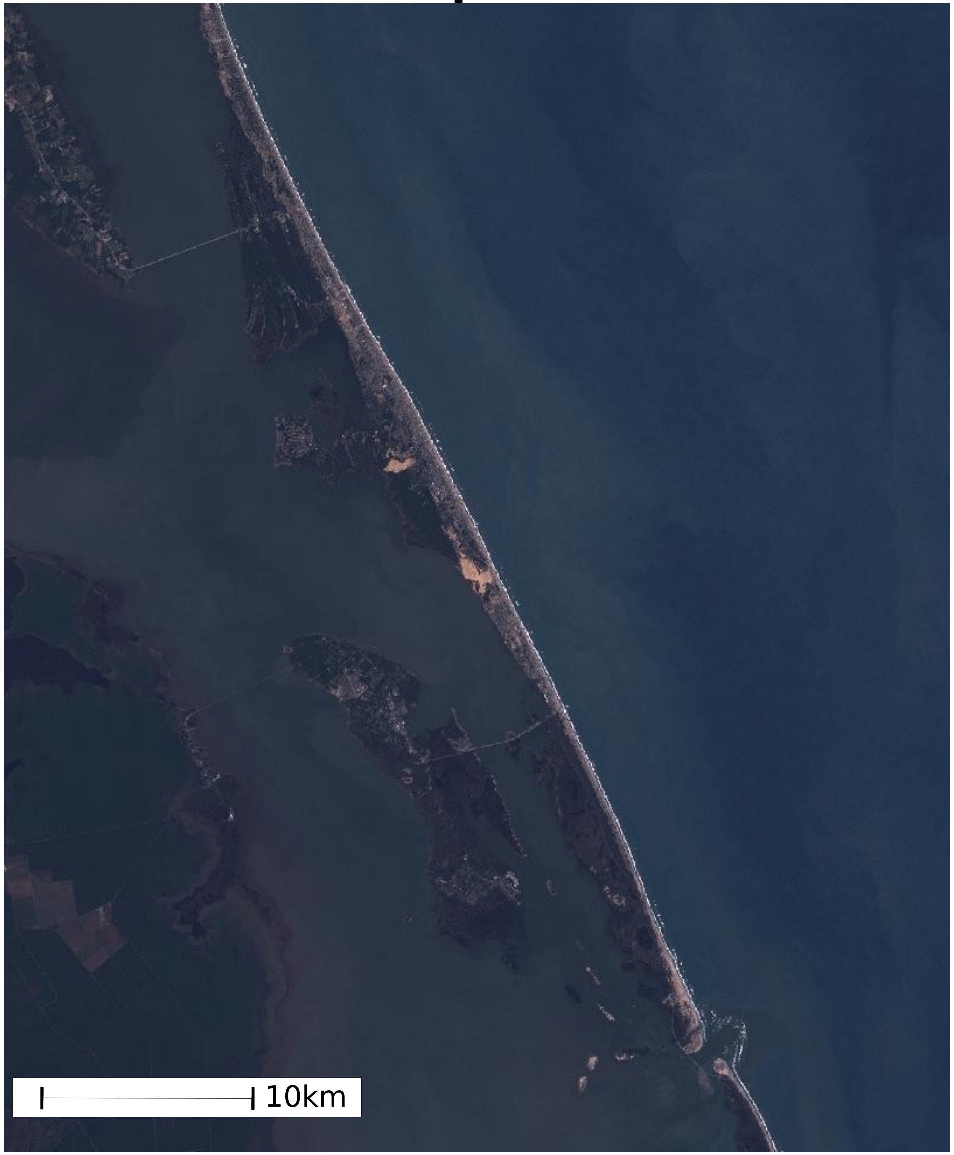
This color composite can be viewed in QGIS (enable GRASS Plugin, load GRASS raster map) or with:

```
d.rast landsat.rgb
```

Visualization of color composites

Color enhancement on larger region

a) unenhanced color composite



Visualization of color composites

Color enhancement on larger region

b) enhanced color composite

Roads vector map overlay

Preparation of vector map

On the liveCD, the road SHAPE file map is in:
$HOME/workshop_material/GRASS-imagery/nc_origmaps/roads/

Get projection info:

```
ogrinfo -so 91479899.shp 91479899
```

Check GRASS location projection info:

```
g.proj -we
```

Reproject SHAPE to UTM18N (WARNING: strange parameter order in 'ogr2ogr ... OUTMAP INMAP'). Enter in one line:

```
ogr2ogr -t_srs "`g.proj -wef`" \
        /tmp/bts_roads_UTM18.shp 91479899.shp
```

Import reprojected map into GRASS:

```
v.in.ogr /tmp/bts_roads_UTM18.shp out=roads
d.vect -c roads
```

Aerial infrared photo import and overlay

Aerial image 71037545.tif was downloaded from USGS (http://nationalmap.gov/ or NC-One site)

Check original projection:

```
cd $HOME/workshop_material/GRASS-imagery/nc_origmaps/aerial/
gdalinfo 71037545.tif
```

Reproject to current location projection (UTM 18N):
Note: -t_srs is *target spatial ref. system*; -tr is *target resolution*

```
gdalwarp -t_srs "`g.proj -wef`" \
         -tr 1 1 71037545.tif aerial1998_UTM18.tif
```

Import into GRASS:

```
r.in.gdal aerial1998_UTM18.tif out=airph98
```

Aerial infrared photo import and overlay

Set current region/resolution to one of the input maps:

```
g.region rast=airph98.blue -p
```

Save RGB composite to new map and display:

```
r.composite b=airph98.blue g=airph98.green \
            r=airph98.red out=airph98.rgb
d.erase
d.rast airph98.rgb
```

Extend region relatively to current region to overlay to LANDSAT:

```
g.region n=n+1000 s=s-1000 w=w-1000 e=e+1000 -p
d.erase
d.his i=nn10 h=airph98.rgb
```

... or do this in QGIS with raster transparency.

Result: Aerial infrared photo overlayed to LANDSAT + Roads

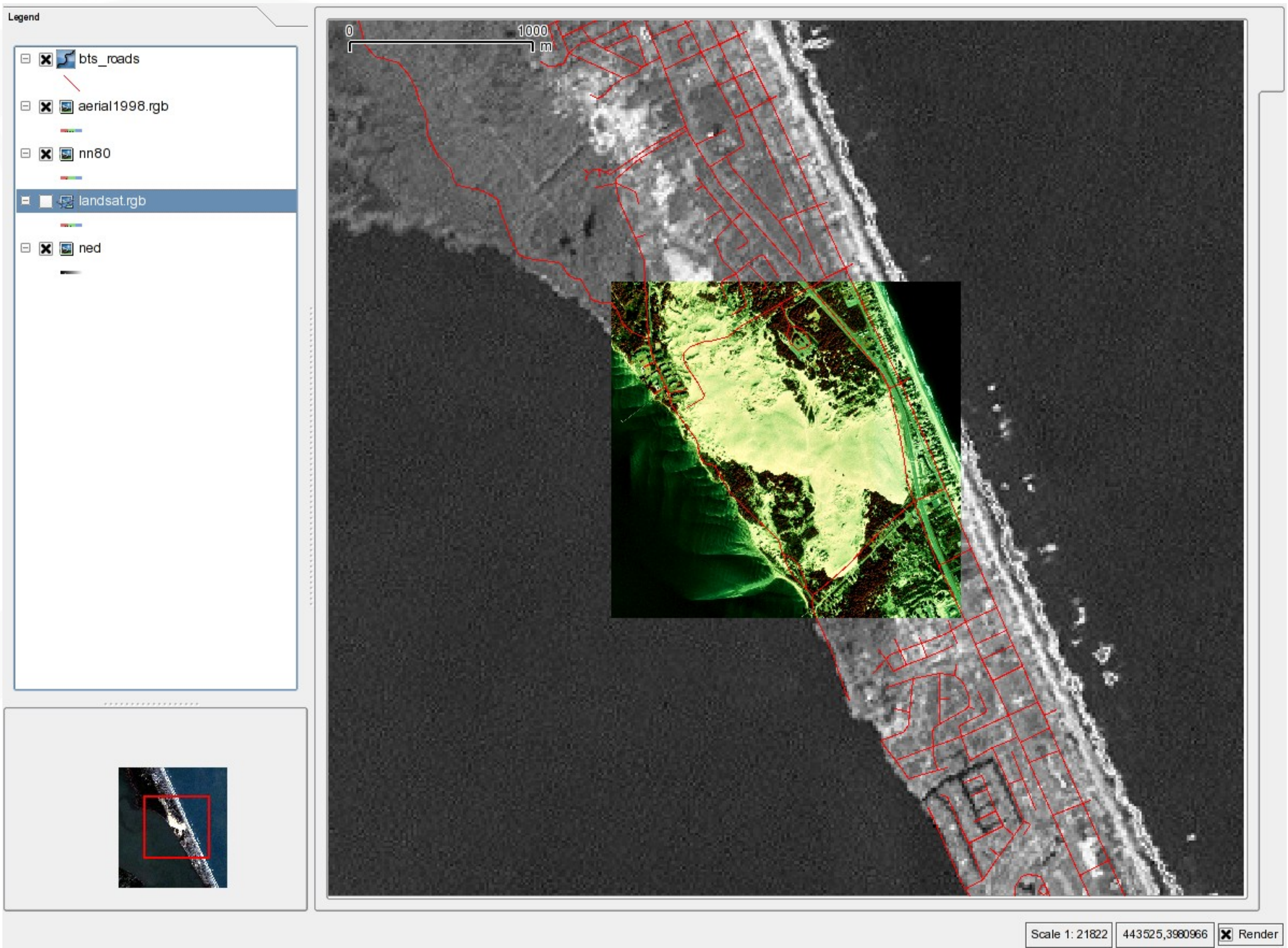


Image fusion

Idea: combine color information with higher geometric resolution from PAN

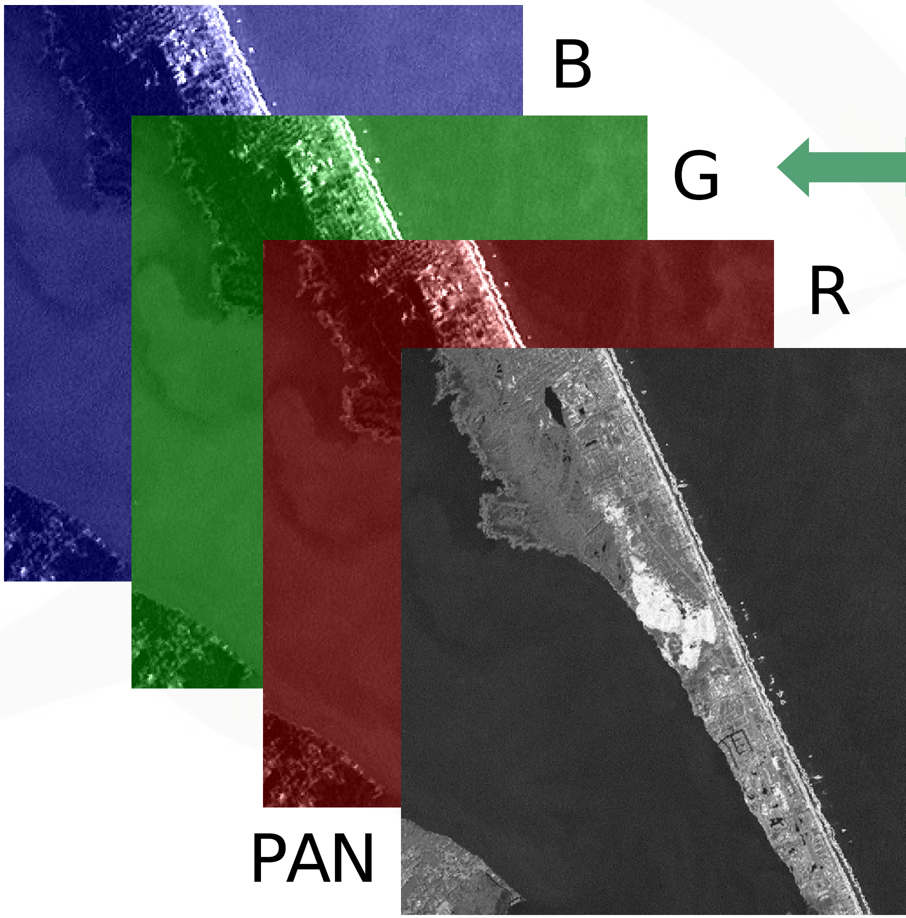


Image fusion: Merge color and panchromatic channels

Brovey fusion (-l for LANDSAT sensor):

```
i.fusion.brovey -l ms1=nn10 ms2=nn20 ms3=nn30 \
                pan=nn80 out=lsat.fusion
g.region -p rast=lsat.fusion.red
d.erase
```

For visualization purposes, we swap assignment to blue and green color “gun” of d.rgb:

```
d.rgb r=lsat.fusion.red b=lsat.fusion.green \
      g=lsat.fusion.blue
d.vect roads col=yellow

r.composite r=lsat.fusion.red b=lsat.fusion.green \
            g=lsat.fusion.blue out=lsat.fusion.rgb
```

Perspective view ('ned' is the DEM):

```
nviz elev=ned col=lsat.fusion.rgb vect=roads
```

Image classification 1/5

Set current region to aerial image:

```
g.region rast=airph98.rgb -p
d.erase
d.rast airph98.rgb

g.list group
```

We add a subgroup from all three RGB channels, this modifies the existing group:

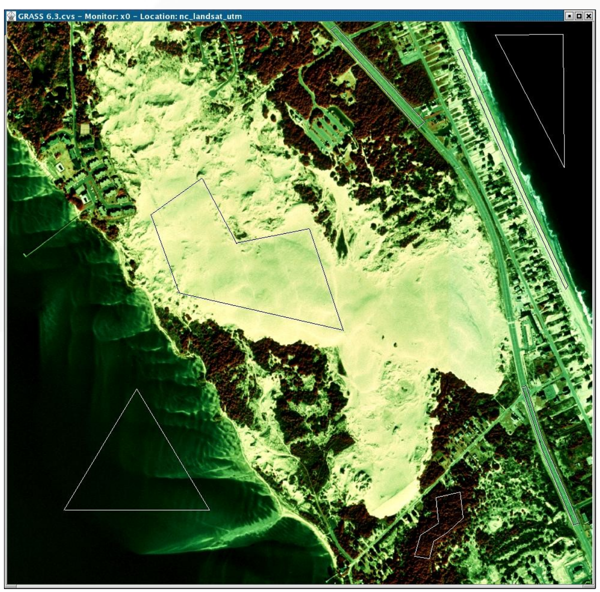
```
i.group group=airph98 subgroup=airph98 \
        in=airph98.blue,airph98.green,airph98.red
```

Image classification 2/5

Digitizing of training areas (not too large areas...) - make the GRASS monitor size as big as possible.

```
# define 'A' areas, use 'x' to save-quit:
r.digit airph98.train
d.rast.leg airph98.train
```

Note that v.digit is much more appropriate, or the ditigizing tool in QGIS.



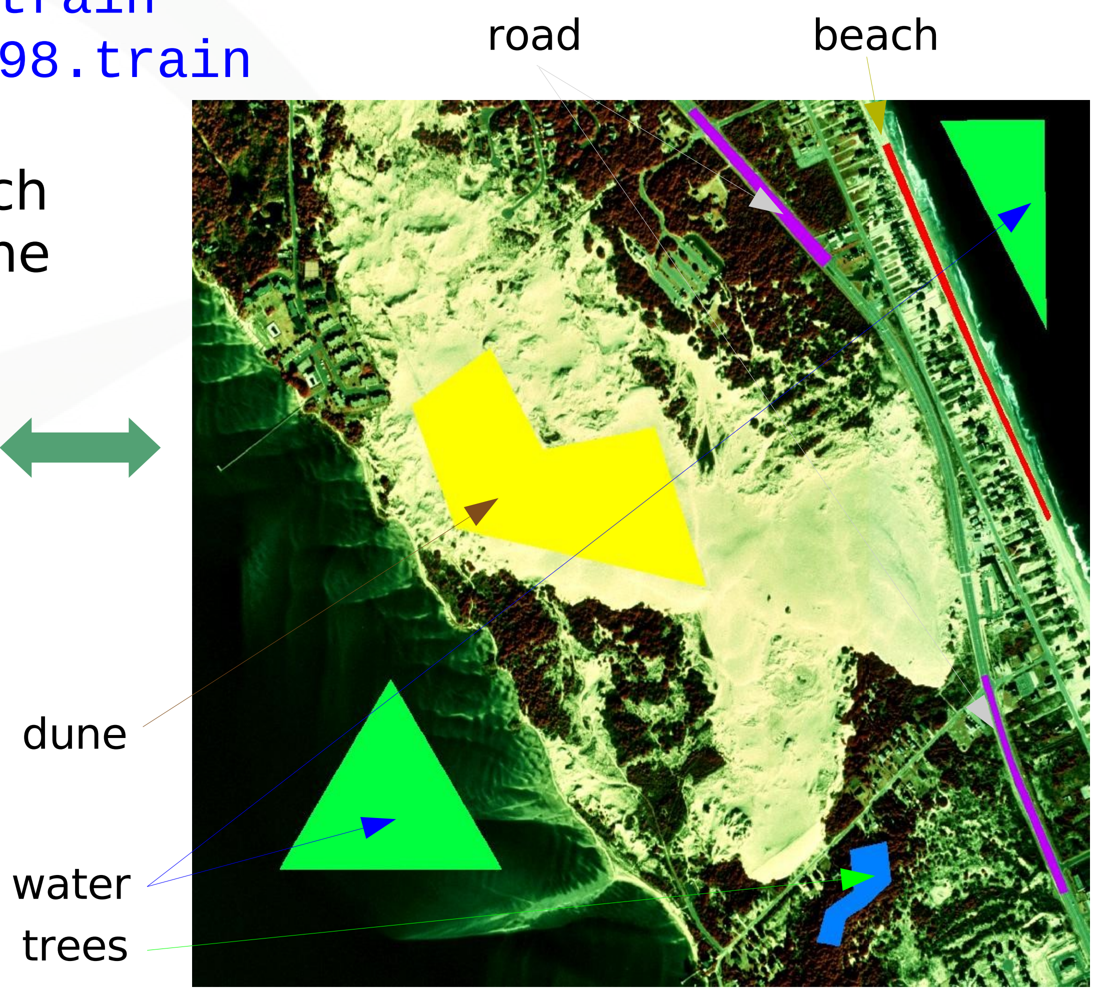


Image classification 3/5

Generating signatures from training areas:

```
i.gensigset airph98.train group=airph98 \
            subgroup=airph98 signature=airph98
```

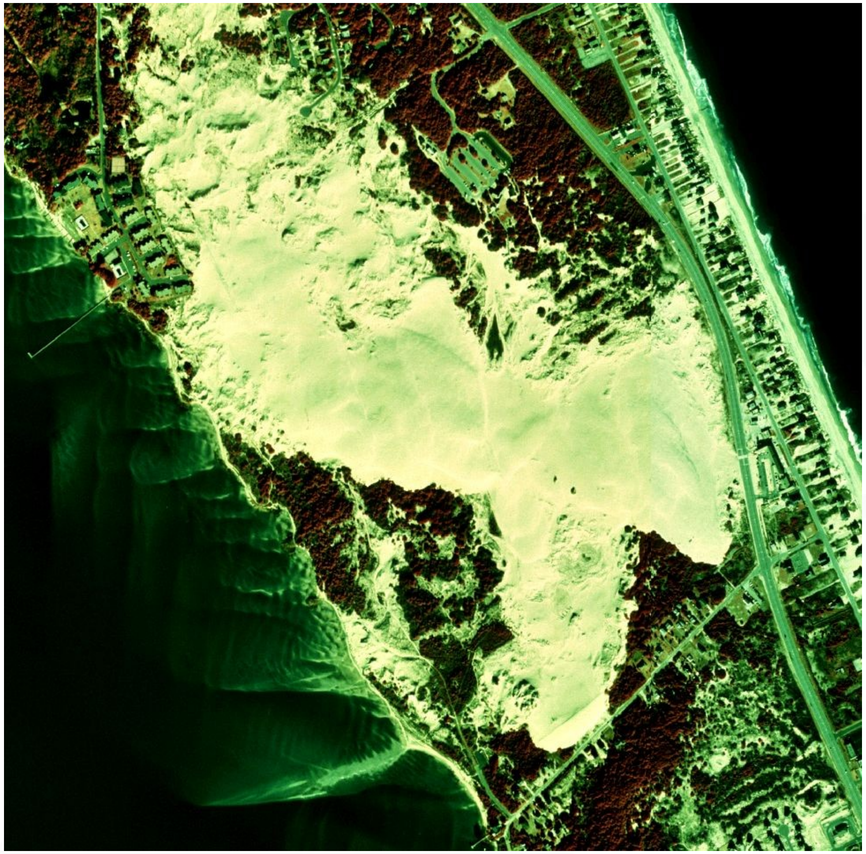
Assign pixels to classes using combined segmentation/radio-metric approach:

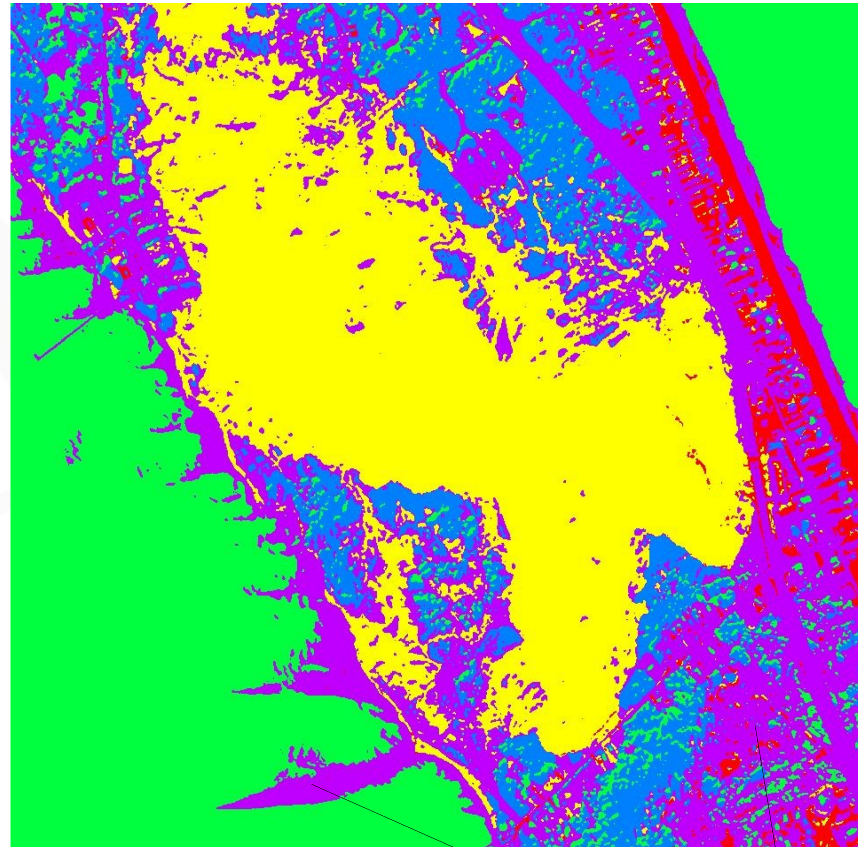
```
i.smap airph98.train group=airph98 subgroup=airph98 \
             signature=airph98  output=airph98.smap
d.rast.leg airph98.smap
```

Filter tiny areas away (5x5 is 25m^2, take dominant class in moving window):

```
r.neighbors airph98.smap out=airph98.smap.filt \
            method=mode size=5
d.rast.leg airph98.smap.filt
```

Image classification 4/5





Remarks:

- not bad for such a simple approach (R,G,B only)
- Needs fine-tuning for sand-in-water, shadows assigned to “road”

    => *See next page for suggestions*

Image classification 5/5

Vectorize resulting polygons of landuse/landcover:

```
r.to.vect airph98.smap.filt out=airph98_smap \
          feature=area
d.rast airph98.rgb
d.vect airph98_smap type=area
```

Extract “dune” and “beach” areas:

```
v.extract airph98_smap out=sand where=”value=1”
d.erase ; d.vect sand type=area
```

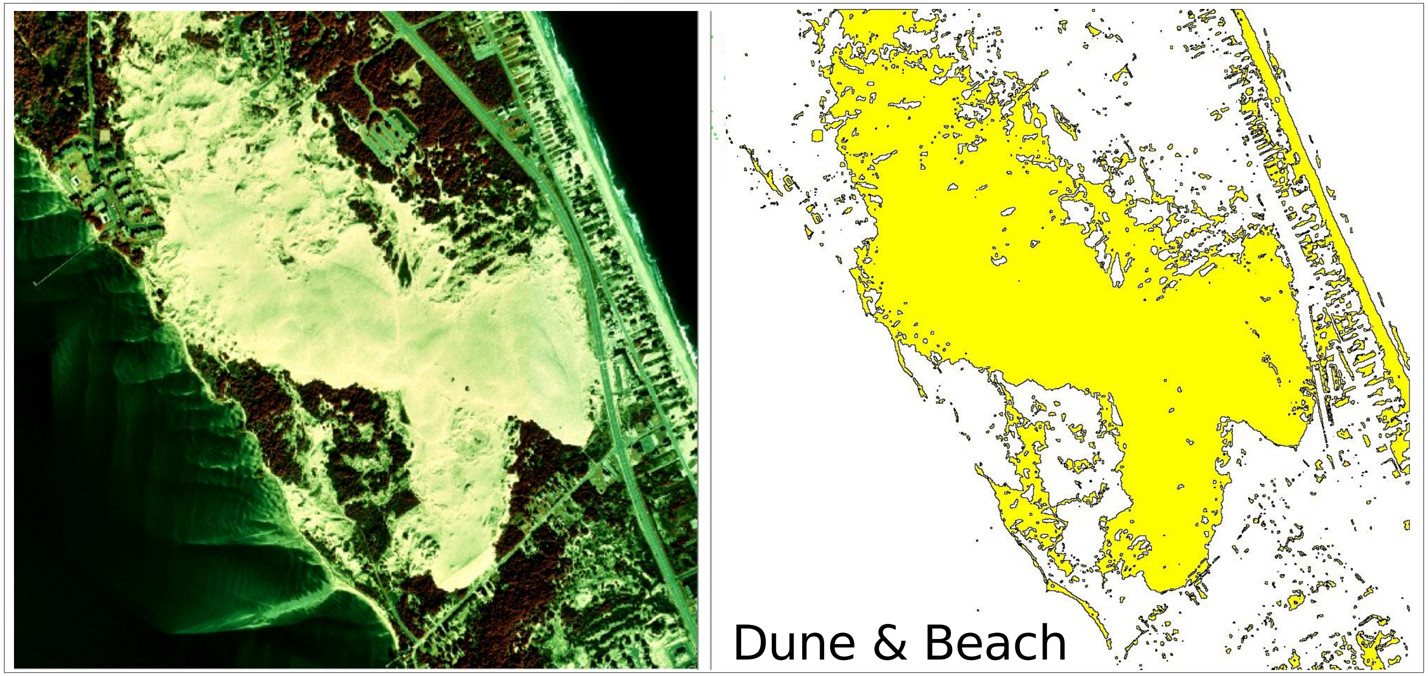


Image classification: Suggestions

Improving the classification results

There are several options to improve the classification of aerial photos or satellite images:

- r.texture can be used to create additional *synthetic* channels based on different textures which are added to the image group
- r.mapcalc can generate ratios of channels (NDVI, EVI, SAVI, ...)
- i.pca can reduce the number of channels (variance reduction), the resulting maps can be used as additional input maps

This requires to run i.gensigset and i.smap again.

Other available tools

What else is possible in GRASS?

- Geocoding of imagery data
  - unreferenced **scanned maps** by defining four corner points
  - **unreferenced satellite data** with set of ground control points
  - **orthophoto** creation using a DEM
  - **digital handheld camera** geocoding
- Calibration of thermal channel (e.g., LANDSAT, MODIS)
- Kappa statistic to verify image classification
- Time series processing

References

- http://grass.itc.it/grass63/manuals/html63_user/imageryintro.html
- `g.manual imageryintro`

Questions? Ideas? Answers?

Join the GRASS community....
http://grass.itc.it

OSGeo.org
The Open Source
Geospatial Foundation

License of this document