Efficient processing of dense point clouds in GRASS GIS

at US-IALE 2016 Annual Meeting

Vaclav Petras (Vashek)

Douglas Newcomb, Helena Mitasova

Center for Geospatial Analytics

NC STATE UNIVERSITY

March 18, 2016

Acknowledgements

Software

Presented functionality is work done by Vaclav Petras, Markus Metz, and the GRASS development team.

Thanks to users for feedback and testing, especially to Douglas Newcomb, Helena Mitasova, Markus Neteler, Laura Belica, and William Hargrove.

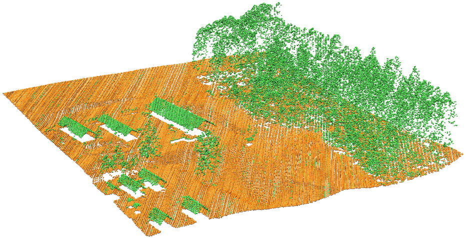
GRASS GIS

Acknowledgements

Datasets

Data for GIS595/MEA792: UAV/lidar Data Analytics course

Nantahala NF, NC: Forest Leaf Structure, Terrain and Hydrophysiology: Lidar data acquisition and processing completed by the National Center for Airborne Laser Mapping (NCALM). NCALM funding provided by NSF's Division of Earth Sciences, Instrumentation and Facilities Program. EAR-1043051. Obtained from OpenTopography.
`http://dx.doi.org/10.5069/G9HT2M76`



Points

- collected by lidar
- Structure from Motion (SfM) from UAV imagery
- a lot of points
    - often more than we need

surface interpolated from points from SfM

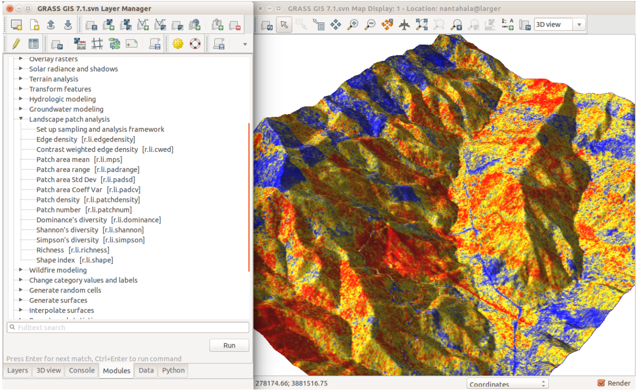
- driven by needs of users
- re-usable
- review
- usable
  - by other people
  - by future myself
- longevity
  - learn now, use forever
  - GRASS GIS over 30 years of development

Open Science Logo, Greg Emmerich, CC-BY-SA-2.0

GRASS GIS

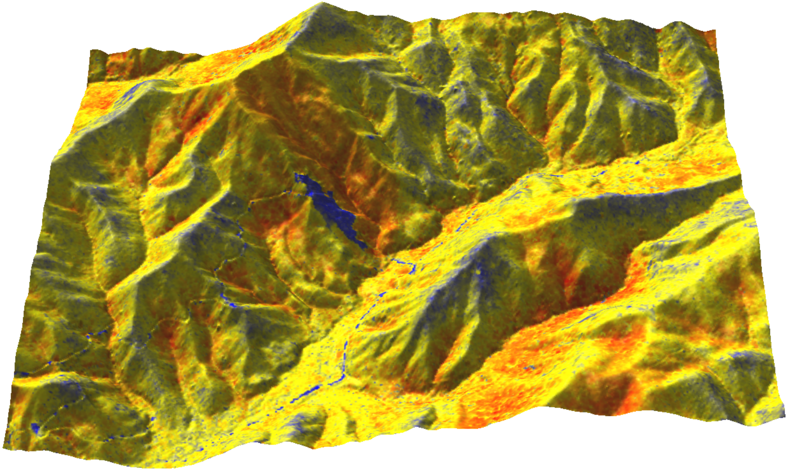
- universal scientific and processing platform
  - GUI, CLI, Python API
  - from small laptops to supercomputers
- lidar processing included
- data size and type challenges

GRASS GIS



Workflow overview

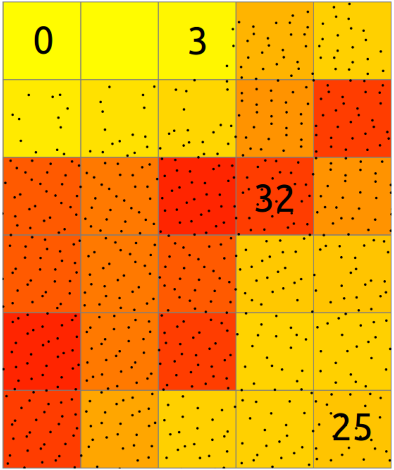
- points → binning → raster → analysis
- points ⟶ decimation ⟶ vector
  - analysis
  - interpolation → raster → analysis



range from *r.in.lidar* on ground obtained from *v.in.lidar* followed by *v.surf.rst*

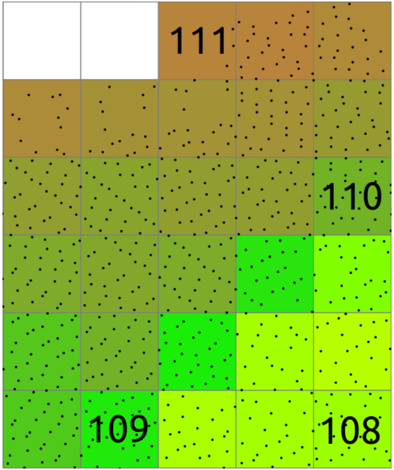
Binning as 2D histogram

- number of points in cell



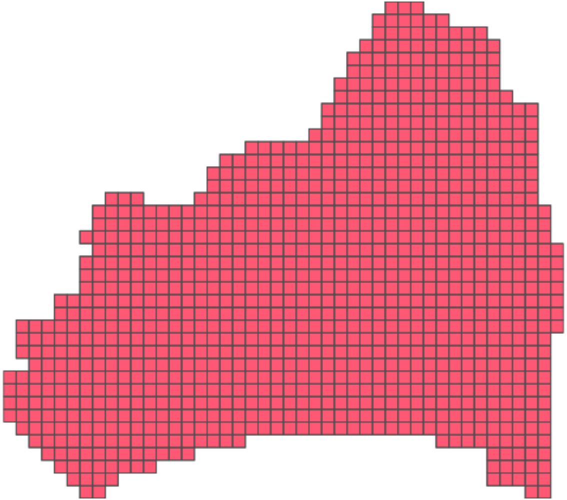
Binning points to raster

- *r.in.lidar* (import and analysis)
- statistics of point counts, height and intensity
  - n, min, max, sum
  - mean, range, skewness, ...



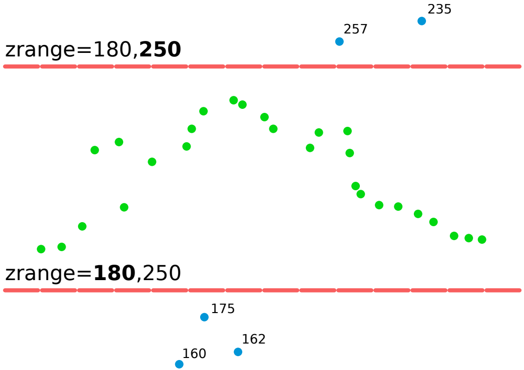
r.in.lidar

- analytical and practical functions
  - multiple tiles at once



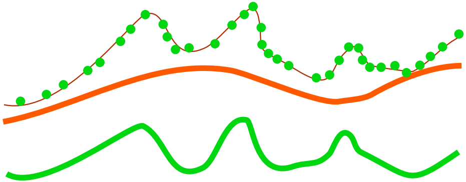
Filtering points

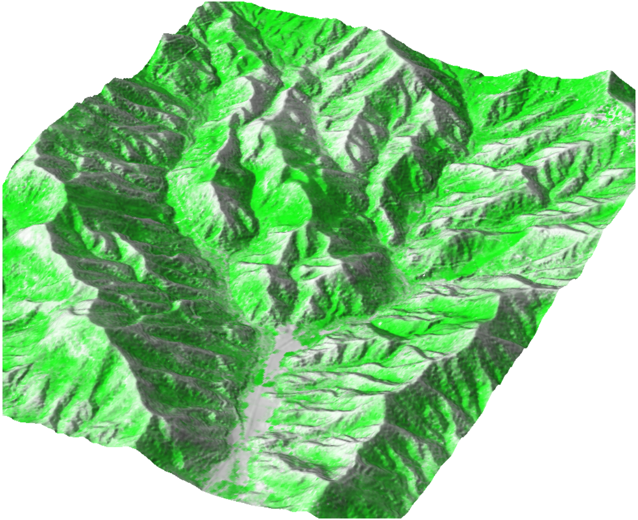
- filter points by
  - range of Z
  - return
  - class
  - ...
- at the time of binning with *r.in.lidar*



Height above a surface

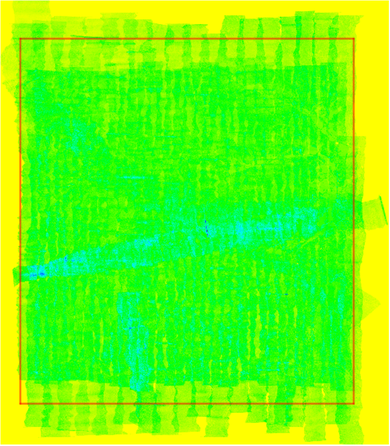
- new feature in *r.in.lidar*
- given surface + points cloud
  $\longrightarrow$ height of features





Rasterize early

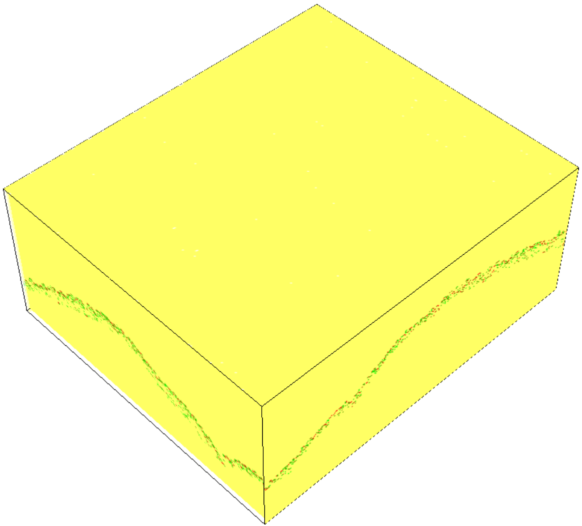
- less cells then points
  - 578 mil points (ground 30 mil)
  - 15 mil cells in 8km $\times$ 7km at resolution 2m
    - faster to loop through
- raster
  - natural spatial index
  - that's what the algorithms use



point density (all classes) at 5m, green $\sim$ 200 points

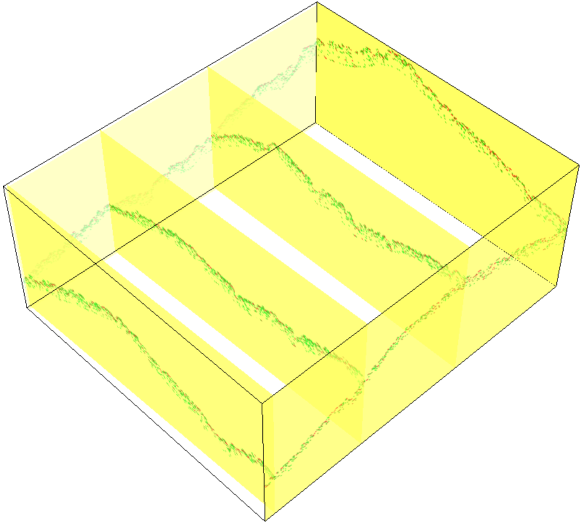
3D raster

- stacked 2D rasters
- challenging to visualize
- same principles as in 2D
  - e.g. 3D raster map algebra



3D raster

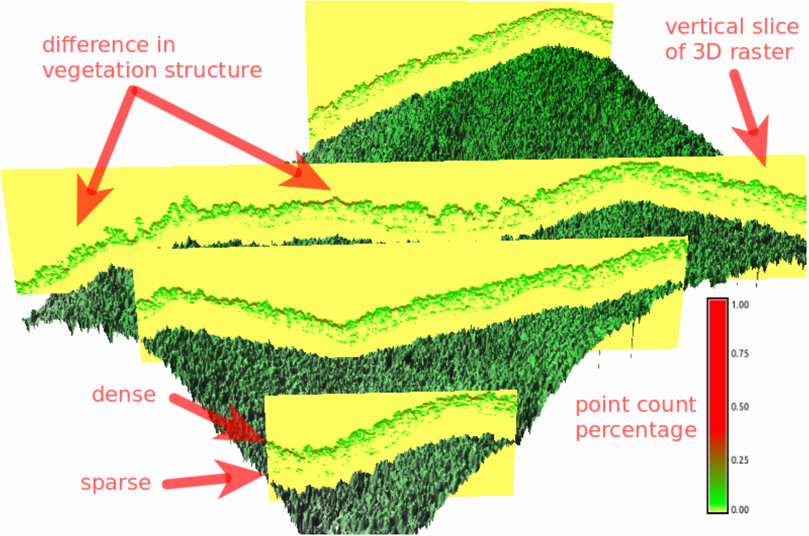
- stacked 2D rasters
- challenging to visualize
- same principles as in 2D
  - e.g. 3D raster map algebra



Binning points to 3D raster

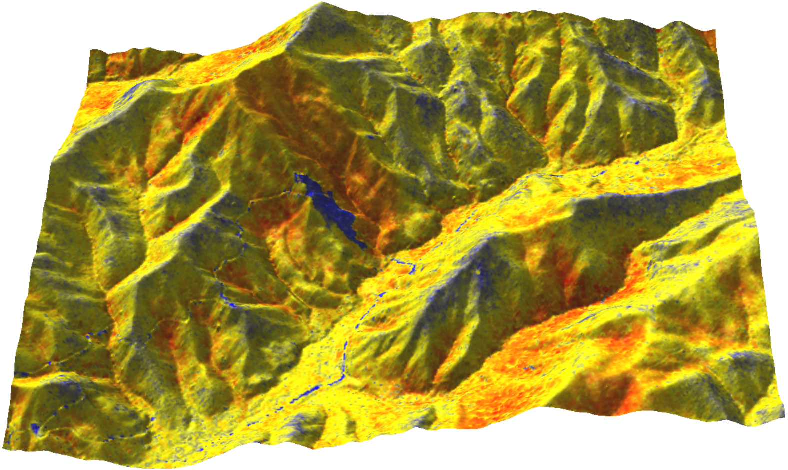
- *r3.in.lidar*
- proportional count
  - count per 3D cell relative to the count per vertical column
- intensity can be used instead of count

under development



Decimation, sampling, interpolation

- *v.in.lidar*
- filtering same as in *r.in.lidar*
- followed by interpolation (costly)
- decimation ≈ sampling
  - fast count-based as effective as more advanced decimation



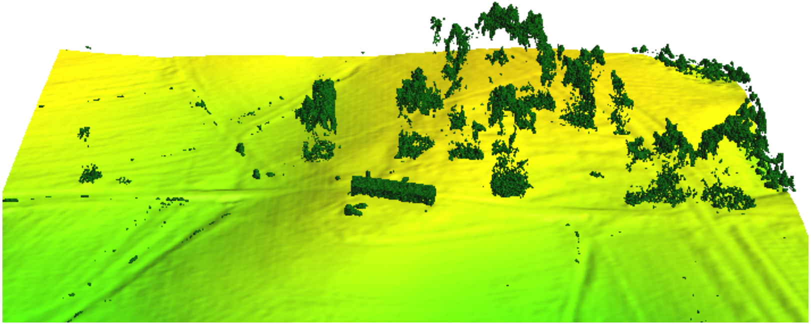
range from *r.in.lidar* on ground obtained from *v.in.lidar* followed by *v.surf.rst*

Large point clouds

- trade-offs: a lot of memory (RAM) or slow
- 64bit version
- point cloud specific optimizations
  - no IDs stored
  - no attribute table
  - no topology created

Ground detection

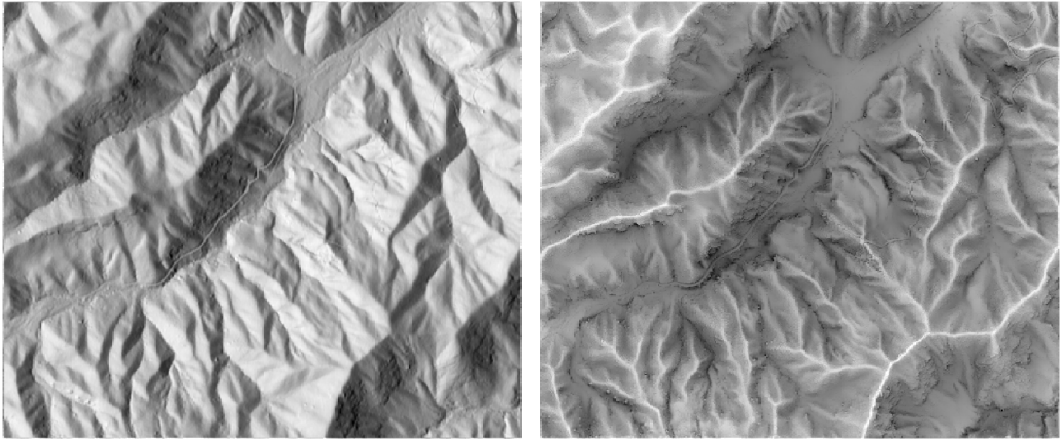
- *v.lidar.edgedetection*, *v.lidar.growing*, *v.lidar.correction*
  - uses returns
- *v.lidar.mcc*
  - multiscale curvature based classification algorithm[1]



[1] Evans, J. S. & Hudak, A. T. 2007: A Multiscale Curvature Algorithm for Classifying Discrete Return LiDAR in Forested Environments.

Sky-view factor

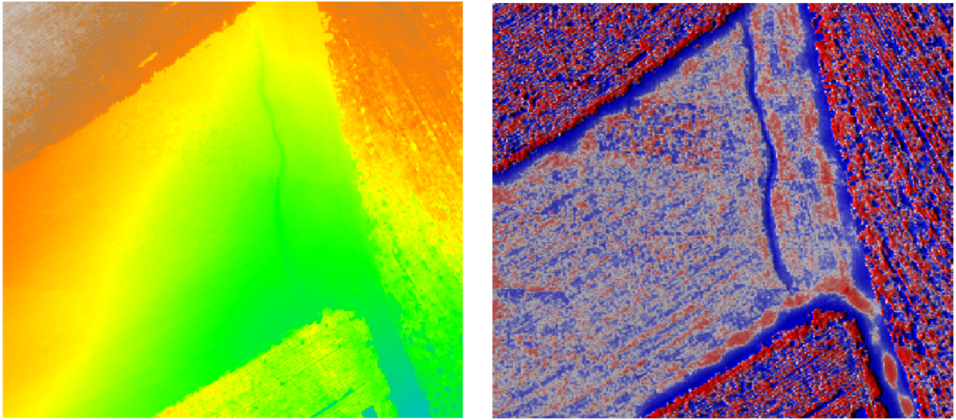
- *r.skyview* (percentage of visible sky)



comparison of shaded relief and sky-view factor

Local relief model (LRM)

- *r.local.relief* (micro-topography, features other than trend)

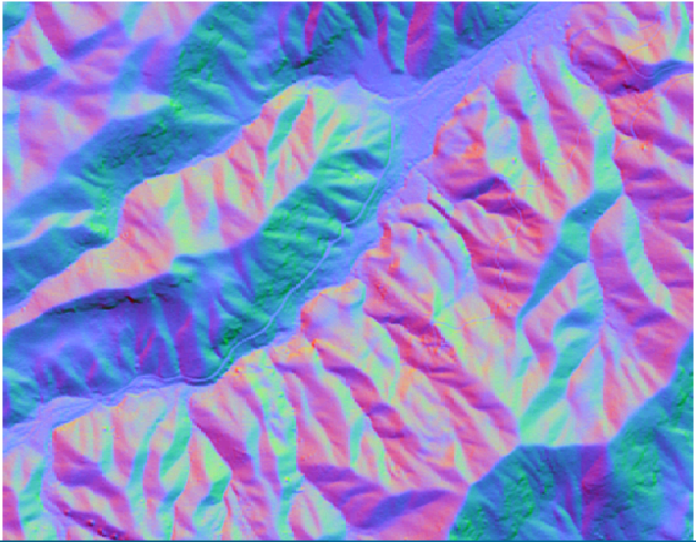


30-60cm wide, 30cm deep, 60m long gully (resolution 30cm)

Analytical shading

- *r.shaded.pca*
- relief shades from various directions
- PCA of shades
- combined into RGB composition

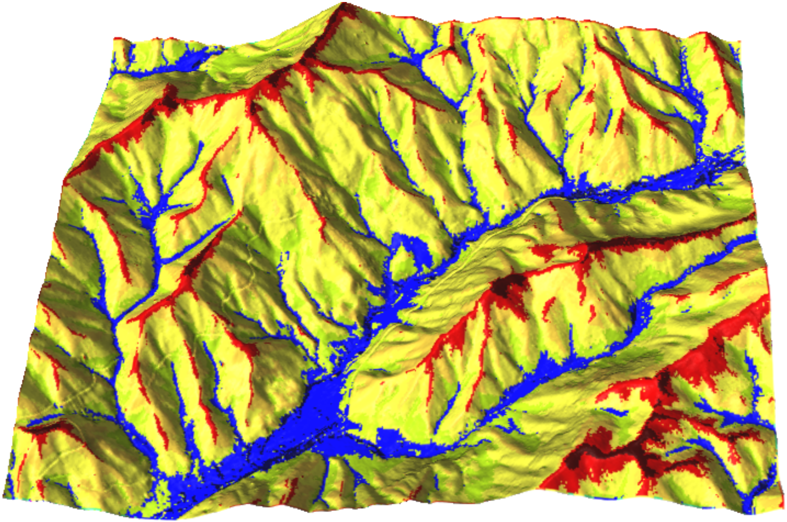
Devereux, B. J., Amable, G. S., & Crow, P. P. (2008). Visualisation of LiDAR terrain models for archaeological feature detection. Antiquity, 82(316), 470-479.



Landforms

- *r.geomorphon*
- geomorphons - a new approach to classification of landform[1]

[1] Jasiewicz, J., Stepinski, T., 2013, Geomorphons - a pattern recognition approach to classification and mapping of landforms, Geomorphology



Integration with PDAL

PDAL

- formats besides LAS/LAZ
- algorithms, filters, decimations

Experimental integration

- *v.in.pdal*
- reprojection during import
- ground filter
- compute height as a difference from ground

Summary

- rasterize early
- make use of existing methods for raster and vector processing
- 3D rasters, PDAL integration
- the plan for next 30 years driven by users – grass-user mailing list

GRASS GIS

Get GRASS GIS 7.1 development version at
`grass.osgeo.org/download`

Slides available at
`wenzeslaus.github.io/grass-lidar-talks`

Paper in preparation: *Processing UAV and lidar point clouds in GRASS GIS*