



GRASS/ OSGeo-News

Open Source GIS and Remote Sensing information

Volume 4, December 2006

Editorial

by Tyler Mitchell

Dear GRASS and other open source users,

It is my pleasure to introduce the future expansion of GRASS-News. The next edition will be morphed into a broader Open Source Geospatial Newsletter covering projects from the OSGeo Foundation and beyond. The aim is to bring relevant news and articles to a larger audience by widening the focus and changing the name.

GRASS-News has always covered more than just GRASS. It has also shown itself to be a high quality production that we can be proud of reading or even printing to share with friends. With the potential of

reaching a larger audience, as well as maintain a single high quality newsletter, the editors have decided to re-brand it as an OSGeo-News production. I have always been encouraged by the content, layout and ideas represented by the GRASS-News volumes and hope to see it continue.

Call For Articles

You can help by providing articles or papers for publication. The next edition of this newsletter will have articles that cover several other open source projects in various topics:

- OSGeo-related news items
- ... continues on next page ...

Contents of this volume:

Editorial	1	Simultaneous simulation of hydrological and carbon cycle processes in a GIS framework	13
GRASS-News Editorial	2	r.roughness – a new tool for morphometric analysis in GRASS	17
FOSS4G 2006 Conference: The meeting of the tribes	2	Resampling SRTM 03"-data with kriging	20
Report on OSGeo Promotions at GIS-IDEAS 2006	3	Interview with Michael Barton	25
Quantum GIS	4	The GRASS Development Team announces GRASS GIS 6.2.0	29
The GRASS user-map	7	The GRASS Development Team announces GRASS GIS 6.2.1	33
Linking GRASS with Chameleon	9		

- GIS
- Remote Sensing
- Web-based mapping and GIS
- Geostatistics / spatial statistics
- Geospatial data access
- and more...

Do you have layout and design skills? You can help by applying your skills to make OSGeo-News even better. Translators into different languages are also needed. You can help by volunteering as:

- Editor
- Layout Designer
- Reviewer

- Translator
- etc.

Please take a moment to thank the editorial staff and contributors of GRASS-News who have made a great newsletter. I am encouraged to see their interest in making the publication applicable to even more people. I look forward to seeing the next edition.

Tyler Mitchell

Tyler Mitchell
 Executive Director
 OSGeo
<http://osgeo.org>
tmitchell AT osgeo.org

GRASS-News Editorial

by the GRASS-News staff

Dear former GRASS-News readers,

As you may have noticed from the joint logo and title on the front page, this is the last issue of GRASS-News in its current format. GRASS is currently in the incubation process as an OSGeo member project, and as part of the integration with OSGeo we've agreed with the OSGeo publicity committee to broaden the scope of the newsletter to cover other OSGeo projects. Many articles have already covered use of GRASS along with utilities from other OSGeo projects, and we're confident that this widen-

ing of coverage can only add to the usefulness and relevance of the articles in the GRASS-News. As a result, from the next issue onwards the newsletter will be known as OSGeo-News. Rather than have less GRASS-specific content, we anticipate the GRASS content will remain the same, and the frequency and/or size of the newsletter will increase to accommodate wider OSGeo material! To this end we need your contributions!

Thank you for your interest in the past and looking forward to OSGeo-News volume 1 with some very interesting GRASS articles included!

Your GRASS-News staff

FOSS4G 2006 Conference: The meeting of the tribes

by Helena Mitsova

The Free and Open Source Software for Geoinformatics (FOSS4G, <http://www.fooss4g2006.org>) Conference held in Lausanne, Switzerland in September 2006 was a great success. It was the first joint con-

ference that brought together the different FOSS4G "tribes", merging the GRASS Users conference series, with the Mapserver, EOGeo and Java developers meetings. Hundreds of people, including software developers, power users and newcomers, had a chance to meet, discuss projects and present new developments and ideas. More than 25 hands-on workshops were offered providing an opportunity for the

participants to learn established and new software tools. Five sold out GRASS workshops were offered including advanced topics such as visualization and lidar data processing. In addition to the plenary sessions, more than 120 presentations were given in eight parallel sessions covering wide range of topics and projects, from Webmapping and desktop GIS to Geodata and OGC standards. The GRASS sessions included presentations on hydrologic modeling, urban planning applications, processing of massive data, support for hazard and emergency response management, visualization, python tools and many others. Jim Westervelt from USA CERL discussed the evolution of GRASS from public domain to FOSS4G in his keynote. All workshop material and presentation slides are available online. Videos and conference photos from the conference are also available on the FOSS4G website under "Souvenirs". Several informal brainstorming sessions were organized as "birds-of-a-feather" (BOF) meetings to work out open access to geospatial data issues, seed national user groups, start new programming projects and more.

During the final afternoon several keynote talks were given as well as a discussion panel on FOSS4G

"politics". Also the second annual Sol Katz award for important contributions of an open source GIS community member was presented. This year Markus Neteler from the Center for Scientific and Technological Research (ITC-irst, Italy) was honored, recognizing his instrumental work on the GRASS GIS project. Congratulations to Markus for the well deserved award thanks for all the work that he has done for GRASS, keeping it growing and getting better for the past 10 years.

The excellent conference organization was carried out by Camptocamp, the University of Lausanne, the Swiss Federal Institute of Technology (EPFL) and the University of Applied Sciences, Western Switzerland. The conference attendance by far exceeded all previous meetings. Great social events accompanied the technical parts of the conference.

The recently established Open Source Geospatial Foundation (OSGeo, <http://www.osgeo.org>) was introduced during the conference and offered an information booth as a focal point for learning more about the foundation and how to get involved. The foundation will take lead on future meetings in the FOSS4G series; the upcoming **FOSS4G2007** will be held in Victoria, Canada from **24-27 September 2007**.

Report on OSGeo Promotions at GIS-IDEAS 2006

8-11 November, Ho Chi Minh City, Vietnam

by Ho Dinh Duan and Venkatesh Raghavan

The GeoInformatics for Spatial-Infrastructure Development in Earth & Allied Sciences (GIS-IDEAS) 2006 Symposium was held in Ho Chi Minh City (HCMC), Vietnam, 9-11 November 2006 (<http://wgrass.media.osaka-cu.ac.jp/gisideas06/>).

The symposium was organised by Japan-Vietnam Geoinformatics Consortium (JVGC) and Institute for Environment and Resources (IER-HCMC) with support from several other institutions and GIS Development as the Media Partner. GIS-IDEAS 2006 Symposium was the third international GIS-IDEAS Symposium to be held in Vietnam since 2002. The main theme of the GIS-IDEAS 2006 was **Geoinformatics for Regional Sustainable Development**. The GIS-IDEAS 2006 Symposium provided a unique opportunity for sharing of knowledge and valuable experiences amongst the academia, researchers, industry and students. The GIS-IDEAS 2006 attracted 150 participants belonging to organisations spread over 9 countries Australia (1), (Belgium (2), France

(1) Germany (5), India (1), Japan (24), Russia (1), Thailand (2), Vietnam(110)).

As a part of activities to increase awareness about Open Source Software for Geoinformatics and promote OSGeo in Vietnam, a pre-conference workshop entitled Open Source for Geoinformatics was organised at the Vietnamese Academy of Science and Technology (VAST) on 8th November, 2006. The details of the workshop **Open Source Geoinformatics Workshop** at the Institute of Information Technology HCMC are as below:

Coordinators: Dr. Tran Van Lang and Dr. Ho Dinh Duan, VAST-HCM by theme Japan-Vietnam Geoinformatics Consortium, the Institute of Information Technology, the Institute of Physics, VAST HCMC and supported by Open Source Geospatial Foundation (OSGeo). 47 participants from Vietnam and 7 participants from Japan attended this workshop.

The pre-conference workshop on Open Source Software for Geoinformatics attracted enthusiastic response with more than 50 participants attending. The workshop provided an overview of GIS applications (Prof. Mamoru Shibayama,



Joint Secretary JVGC) and Open Source software tools (Prof. Venkatesh Raghavan, Director, OSGeo). The afternoon session included actual installation of Open Source software tools and demonstrations including Vietnamese language localisation (Dr. Go YONEZAWA, Kyoto University and Mr. Ninsawat SARAWUT, Osaka City University) Participants were presented with the Osaka City University (OCU) FOSS4G toolkit CD including GIS/RS and Web-GIS software for self-learning.

The post-workshop meeting for establishing the Vietnam after the workshop that was attended by more than 40 participants and members decided to establish the Vietnam OSGeo Chapter. Mr. Dao Van TUYET, Institute of Information Technology, VAST-HCMC was nominated as the Vietnam OSGeo Chapter representative and Dr. Ho Dinh Duan, Institute of Physics, VAST-HCMC was nominated as one of the coordinators of the Vietnam OSGeo Chapter. The proposal for establishment of OSGeo Vietnam Chapter is being prepared by the chapter representative and will be finalised in January, 2007. The highlight of the workshop along with the news about the proposed Vietnam OSGeo Chapter was broadcast on Ho Chin Minh television news on 8th November, 2006. This was per-

haps the first official television news program in the world wherein OSGeo was specifically mentioned several times (http://wgrass.media.osaka-cu.ac.jp/gisideas06/foss4g_ws_on_hcmctv.swf).

The OSGeo booth was operated on all the three days (9-11 November, 2006). The poster session was preceded by lightning-talk on proposed activities of OSGeo Vietnam by Mr. Do Long Van. Details about activities of OSGeo were also presented in the Technical Session by Prof. Venkatesh Raghavan (<http://wgrass.media.osaka-cu.ac.jp/gisideas06/viewabstract.php?id=166>). The OSGeo Booth at GIS-IDEAS 2006 included attracted attention of almost all the symposium participants. Brochure and posters explaining the activities of OSGeo and various OSGeo projects were made available at the booth.

*Dr. Ho Dinh Duan
Institute of Physics
VAST – HCMC, Vietnam*

*Prof. Venkatesh RAGHAVAN
Graduate School of Creative Cities
Osaka City University, Japan*

Quantum GIS

What's New in 0.8

by Gary E. Sherman

Introduction

Version 0.8 of Quantum GIS (QGIS) is nearing release. Many of you may be wondering why its been so long in coming. At 0.8, we converted the QGIS code base to use Qt 4.x. This was a major overhaul and took longer than anticipated. With that job behind us, we look forward to more frequent incremental releases in the future.

In addition to the port to Qt 4.x, version 0.8 adds a number of new features and enhancements.

New Features and Enhancements

New features in QGIS 0.8 include:

- WMS support
- Improved vector and attribute editing
- Improved measure tools with area measuring
- Attribute searching
- New legend structure
- Refactoring of API to allow the use of QGIS libraries in mapping applications
- Improved MapServer export tool

- Vector layer transparency and antialiasing
- GRASS support in all platforms
- Enhanced GRASS support and toolbox commands
- Enhanced vector editing, including copy, cut, paste, snapping and vertex editing
- Shapefile/OGR layer editing

Based on an informal poll on <http://qgis.org>, it appears that people are most interested in the GRASS tools, WMS, MapServer export, and enhanced editing.

GRASS Tools

More tools have been added to the QGIS GRASS toolbox and the interoperability has been enhanced. For those unfamiliar with the QGIS/GRASS integration, the GRASS plugin allows you to:

- Create new GRASS vector layers
- Easily import GDAL/OGR supported layers into GRASS
- Perform many processing operations, including union, intersection, subtraction, buffering, feature extraction, network analysis, create slope and aspect maps, raster to vector conversion, and layer export to other formats.
- GRASS shell gives you access to all GRASS functions from within QGIS
- Edit vector layers and attributes
- Layer browser

Figure 1 shows the new GRASS browser that allows you to view layers and detailed information about each. From the browser, you can add a layer to the map and also copy, rename, or delete a layer.

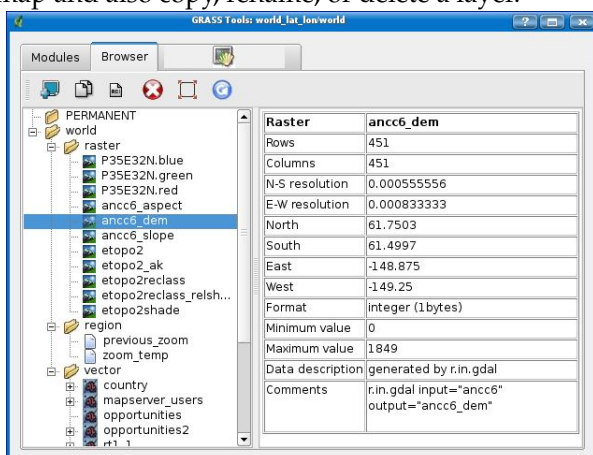


Figure 1: GRASS Browser

The GRASS shell gives you full access to all command line features of GRASS. This allows you to run GRASS commands that aren't implemented in the toolbox.

WMS

WMS support has been added for 0.8. This allows you to add layers from web mapping servers and view them along with your other vector and raster data in QGIS. A default set of servers is provided to help those new to WMS. Figure 2 shows the Server Connections dialog with the QGIS WMS layers displayed and ready to be added to the map canvas.

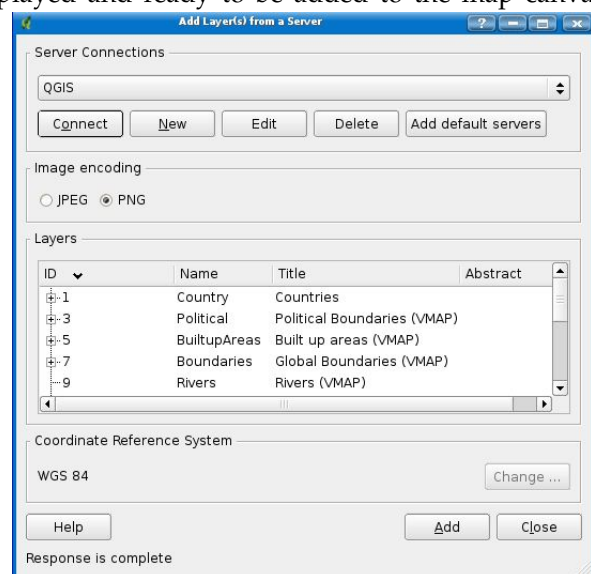


Figure 2: WMS Server Dialog Showing Layers from the QGIS Server

The WMS implementation also supports identifying vector features and a number of image formats.

MapServer Export

The MapServer export tool has been completely revamped at 0.8 and provides full support for class rendering, rasters, projections, and pretty much all the things that were missing in the original version.

The export tool now operates on a saved QGIS project rather than the currently loaded layer set. In addition, it is a standalone binary, allowing you to run it outside QGIS to export a set of project files.

Under the hood, the export tool uses a Python script (`ms_export.py`) to do the work with a GUI interface to collect the required inputs. You can use the Python script without the GUI, allowing you to script the export of a large number of QGIS project files.

Figure 3 shows the GUI front-end for the MapServer export tool. All inputs are optional except for the name of the map file

to create and the QGIS project file to export.

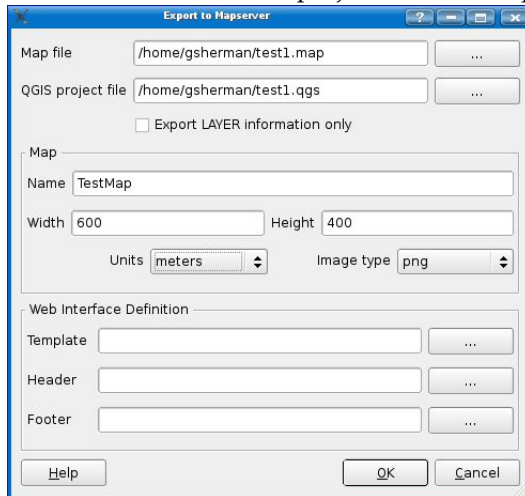


Figure 3: GUI for MapServer Export Tool

To use the script without the GUI, you need to provide all the inputs expected by `ms_export.py`. This can be done with a driver script and an example is included in your QGIS install at `./share/qgis/python/test_export.py`.

The MapServer export tool doesn't currently support fonts, label properties, point symbols, and custom line or polygon symbols. Of course the map file can be easily customized in your favorite text editor after export.

Enhanced Editing

While the QGIS/GRASS plugin already provides advanced editing capability, editing has been significantly improved for shapefiles and PostGIS layers at 0.8. You can now copy/cut/paste features between these layers. Vertex editing, including snapping has been added as well.

Other Highlights

Another new feature is the ability to search the attribute table. Figure 4 shows the new Attribute Table dialog with the search controls at the bottom of the dialog. You can search by any field in the attribute table. The Advanced search provides a complete query builder to facilitate complex searches.

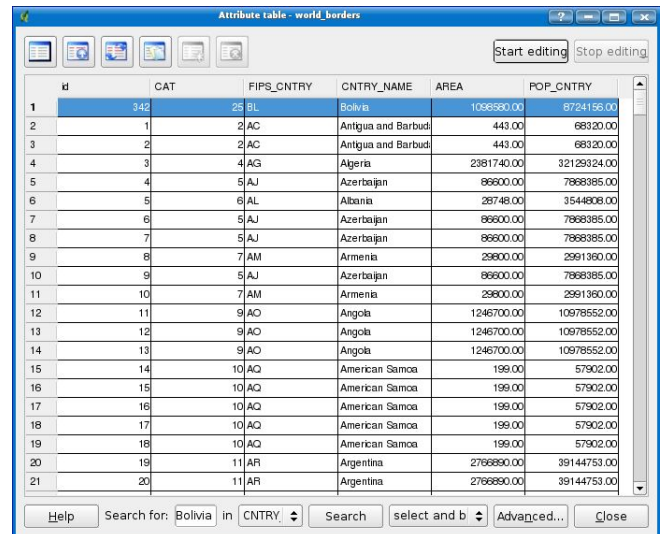


Figure 4: Attribute Searching in QGIS 0.8

The measuring tools have been enhanced and now support the measurement of area in addition to distance.

Native GRASS support is provided for all QGIS platforms, including Windows. This means you don't need Cygwin to run GRASS.

WFS support is included in the 0.8 code base but it is not built by default since it is considered to be potentially unstable. Those of you compiling QGIS from source will have the option to include WFS support.

What's Ahead

There are some exciting developments underway within the QGIS project. Foremost among these is the implementation of Python bindings for the QGIS libraries. This will allow developers to easily develop custom mapping applications in Python and PyQt4. Additional refactoring of the QGIS libraries has already been done and will be merged into the code base after the 0.8 release.

At 0.9 a mature API will be available for developing your own mapping applications in either C++ or Python. We also anticipate the ability to automate QGIS tasks as well as develop plugins using Python.

The remainder of the work leading up to QGIS 1.0 will concentrate on stabilizing and maturing existing functionality, as well as adding new features such as raster image catalogs, improved labeling, and additional data providers.

Gary E. Sherman
Quantum GIS
<http://qgis.org>
sherman AT mrcc com

The GRASS user-map

by Stephan Holl

GRASS User-map

Like nearly all Open Source GIS projects GRASS now also has an usermap. From the very beginning of the usermap in March 2006 it now has about 736 worldwide users marked on it (and the number of points are constantly growing...).

The idea behind this is creating a map with all spatially distributed GRASS users in the world in one map. Furthermore, the GRASS project wants to be collaborative with other GIS-related OSS projects like QGIS, MapBender, MapServer etc. Their usermaps are included using OGC services.

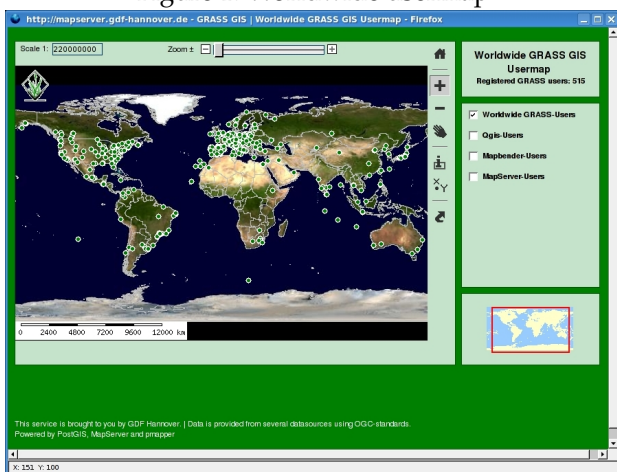
You can reach the usermap using this URL and create yourself a point on the map: <http://maps.gdf-hannover.de/grassusers/>

Technical details

Not surprisingly the usermap is consequently set up with Free software. The basis is built on top of Debian GNU/Linux using PostgreSQL/PostGIS for data storage.

The rendering is done by UMN MapServer while the client is a stripped down mppaper1-client.

Figure 1: Worldwide usermap



Backdrop data is pulled from various OGC services. The following list of data services is currently implemented. If your OSS GIS project is not listed below, feel free to contact us.

- JPL BLue Marble WMS-Worlddata and US Landsat-Satellite data

- PostGIS-based point layer with the GRASS-users.
- QGIS-usermap as WFS
- MapBender-usermap as WMS
- MapServer-usermap as WFS
- State Boundaries from <http://mappinghacks.com/data/> imported in our own PostGIS-DB

How to enter


Marking yourself on the map is very easy. Just use this button  and fill in the form given in fig. 6.

Figure 2: Enter yourself to the map

After choosing 'Insert your point' the map will be updated and your point is shown.

At a deeper zoom stage you can see your point labelled with your name.

GRASS users as OGC WMS/WFS services

As the GRASS usermap makes heavy use of OGC services to pull backdrop maps from other servers, the software offers the GRASS users as WMS/WFS data as well.

The following URLs grab the GetCapabilities from the GRASS user WMS- and WFS-server:

```
# grab WFS GetCapabilities
http://maps.gdf-hannover.de/cgi-bin/
grassuserwfs?REQUEST=GetCapabilities\
&SERVICE=WFS&VERSION=1.0.0
```

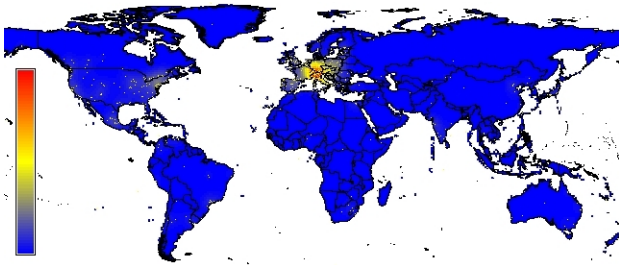
```
# grab WMS GetCapabilities
http://maps.gdf-hannover.de/cgi-bin/
grassuserwms?REQUEST=GetCapabilities\
&SERVICE=WMS&VERSION=1.0.0
```

Therefore it is easy to use the GRASS usermap in your WMS/WFS client of your choice as a separate layer. Additional attribute data is served through a WFS service.

Use the data in GRASS

GRASS GIS does not have a WFS reader itself (it is not difficult to write a script `v.in.wfs` though), but with the following tool-chain you can easily import the GRASS users into your location.

Figure 3: Kernel-density-map of worldwide GRASS-users. The gradient ranges from red over yellow to blue for high to low density. Yellow dots indicate the location of single users.



Import

Note that the WFS-server only offers the LatLon-projection (EPSG:4326) as seen in the GetCapabilities document queries earlier.

Assuming we already have a LatLon-location set up we can import the data very easily. There are only two steps needed to fetch the actual GRASS users from the WFS and use the resulting points as GRASS-vectors:

1. query the WFS servers and save the resulting GML file into a temporary file
2. import the temporary file into GRASS (using `v.in.ogr`)

```
# use curl to get GML from the server
curl -o /tmp/grass_users
"http://maps.gdf-hannover.de/cgi-bin/
grassuserwfs?REQUEST=GetFeature&
Typname=grass_users\
&SERVICE=WFS&VERSION=1.0.0"
```

```
# import into GRASS
v.in.ogr --o in=/tmp/grass_users
out=GRASS_users
```

Note that we need to apply the `-o`-switch since the resulting GML file does not provide useful projection information.

Generate hot-spots of GRASS-users

To show where the GRASS user hot-spots are located we can calculate the kernel-density using the GRASS-module `v.kernel`.

```
# use standard deviation of 1
v.kernel -v in=GRASS_users out=kernel_stdd1
```

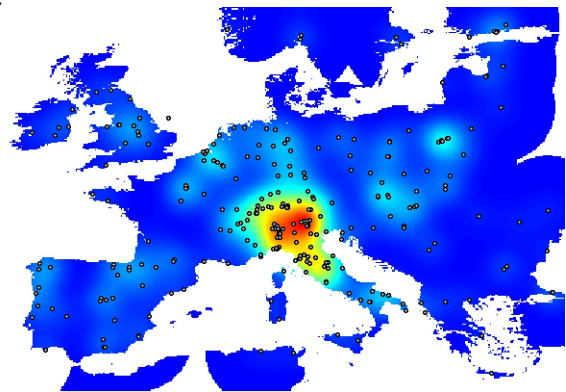
```
stddeviation=1
```

```
# set 0 to NULL
```

```
r.null map=kernel_stdd1 setnull=0
```

This module creates a raster file called `kernel_stdd1`, which is shown in figure 6 and for a subset of Europe in figure 6. For a better visualisation you can add any other dataset you like. Here we have chosen the free world-data available from <http://www.mappinghacks.com/data/admin98.zip>.

Figure 4: Kernel-density-map of GRASS-users in Europe. The gradient ranges from red over yellow to blue for high to low density. Dots indicate a single user.



Conclusion

Having this described in short you can see that GRASS is currently mainly used by Europeans. Feel free to add yourself and play around with the data of the usermap.

Have fun!

Literature and links

- Pmapper <http://www.pmapper.sourceforge.net>
- Debian GNU/Linux <http://www.debian.org>
- PostgreSQL <http://www.postgresql.org>
- PostGIS <http://postgis.refractor.net>
- UMN MapServer <http://mapserver.umn.edu>
- Mapping Hacks <http://www.mappinghacks.com>

Stephan Holl
 Intevation GmbH
 Osnabrück
<http://www.intevation.de>
stephan.holl@intevation.de

Linking GRASS with Chameleon

Geo-analysis on the web

by *Massimiliano Cannata*

Introduction

Nowadays Web-GIS services are one of the most growing sector in the geographical information system science. It's popularity mainly reside in an easy to use interface for non-specialist to access geographical information in order to take decision. In most cases these web applications are focused on distributing geo-spatial information on the Internet in a "static" way: users can access and navigate the different maps, build combination of layer and print the results. More "dynamic" applications are now requested by different specialists: geologists ask for digitalisation tools, hydrologists ask for watershed analysis tools, and so on. Providing this kind of geo-analysis tools requires full access to typical GIS functionality like maps generation, map editing and map analysis. In this paper a new procedure for linking a package for deploying and managing Web mapping applications (Chameleon) and a package for geospatial data management and analysis (GRASS) is shown: such a procedure consist in the development of custom Chameleon's widgets accessing GRASS functionalities. Because of the FOSS environment of both this two softwares, this is a free, transparent and highly customised solution for developing "Web-GIS analysis tool".

Linking GRASS with Chameleon

Chameleon overview

Chameleon ([DM Solution Group](#), 2006) is an highly customisable and adaptable environment for deploying and managing Web mapping applications. By using MapServer ([University of Minnesota](#), 2006) as the back-end mapping engine that generates map images, manages mapped data and handles all of the geographic processing, Chameleon Web Mapping Components (CWC) provide an html-like tag system to incorporate in your current html pages the required mapping functionalities called widgets. In order to build a Web-GIS application with Chameleon the developer needs to handle three files: the initialisation file (lunching the application), the mapserver mapfile (defining the layer's properties), and the Chameleon template (designing the look and the features of the application). This last file is the place where all the desired geographical tools (widgets) have to be inserted within the HTML, by using the defined tags and properties, in order to design

the look of the Web interface. In the following lines a simple chameleon template file is shown as an example:

```
<html>
  <head>
    <title>CHAMELEON TEMPLATE</title>
  </head>
  <body onload="CWC2OnLoadFunction()">
    <form>
      <table>
        <tr>
          <td align="center">
            <CWC2 TYPE="MapDHTML"
              VISIBLE="true"
              WIDTH="400"
              HEIGHT="300"
              ALLOWRESIZE="true"
              MARQUEECOLOR="#FF3333"
              MARQUEEWIDTH="2"
              MINSCALE="1"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

Three elements can be noted : the FORM enabling the user-server interaction, the CWC2OnLoadFunction() enabling the processing of the argument passed by the FORM, and the CWC2 defining a Chameleon widget (in this case the map with the chosen options).

GRASS call requirements

Three facts need to be considered in order to allow users to access GRASS functionality from the web in a safe way:

1. GRASS need a particular environment setting for running: this setting is based on particular environmental variables values and files and is generally set by the starting grass shell script.
2. GRASS allows just one user access for one MAPSET at a time. As GRASS has been designed for a desktop utilisation each MAPSET has his own "active" settings: multiple users could result in changing this values affecting all other running processes (i.e. two user using two different extension parameters for grass analysis).
3. Every web user needs to have his data protected (from other users) and available till the active PHP session is alive.

Technical solution

To handle with these requirements a specific PHP class (grassOBJ) with the following methods has been developed:

- in the constructor most of the object variables are set;
- in the set_mapset_grassrc method a unique MAPSET and a unique .grassrc6 file (containing some GRASS variable) for the active PHP session are generated and set in the corresponding object variables;
- in the set_grass_env method all the system environment variable to be able to run GRASS are set;
- in the set_grass_region method the selected region is set as the active grass region.

A special Chameleon widget called SharedResource (that has no associated event but allows the setting of common variables) is used to specify the general GRASS variables values: an example of its structure is shown in the next lines.

```

1. <!-- Shared Resource for GRASS setting-->
2. <cwc2type="SharedResource" name="GRASSConfig">
3. <gisbase value="/usr/src/grass-6.0.0/bin"/>
4. <gisdbase value="/var/www"/>
5. <location value="NorthAmerica"/>
6. <projection value="epsg:2163"/>
7. <g_errors value="/var/www/msg"/>
8. <g_messages value="/var/www/msg"/>
9. </cwc2>

```

Where: line 2 open and define the sharedresource widget type and name; line 3 define the path to the binary grass commands; line 4 - define the DBASE path; line 5 define the LOCATION name; line 6 define the PROJ (epsg:XXXX) value for the grass LOCATION name (if not listed in the epsg file, just use the GRASS command "g.proj -j" and add the parameters in a new line with a new epsg number); line 7 define the path to the grass log error file; line 8 define the path to the grass log message file; line 9 close the sharedresource widget;

Taking advantage of the grassOBJ class new chameleon's widgets using GRASS functionalities (herein referred to as "GRASS widget") can be developed. Knowing that every widget object has the following base methods:

1. constructor - here the parent constructor is called and the attributes of the specific widget are defined;
2. InitDefaults, initialises the widget with the right values (the object variables are set);
3. ParseURL - reads the posted variables and executes the different functions (the core of the widget, where things happens);
4. DrawPublish - here the widget appearance (button, list, box etc..) is defined;

By adding the GRASSConfig shared resource as a parameter of a new GRASS widget all the needed setting will be available, therefore in the InitDefaults function a new grassOBJ object can be generated. Then in the ParseURL function the GRASS environment variable can be set by using the set_mapset_grassrc and set_grass_env methods. In the same ParseURL function all the mapscript functionalities and the GRASS commands text variable needed can be generated and executed. The following lines show how a GRASS widget calling the r.stats command can be implemented.

```

class GrassStatistics extends CWCWidget {
    //generic variables
    var $moButton;
    var $moPopup;
    var $moLabel;
    var $layer;
    var $filename;

    //variables for Grass
    var $GrassConfigResource;
    var $oGRASS;
    var $config;
    ...
    // constructor
    function GrassStatistics() {
        // invoke constructor of parent
        parent::CWCWidget();
        ...
        $this->maAttributes['GRASSCONFIGRESOURCE'] =
            new StringAttribute('GRASSCONFIGRESOURCE',true);
        ...
    }
    function InitDefaults() {
        ...
        $this->GrassConfigResource =
            $this->maParams['GRASSCONFIGRESOURCE'];
        $config =
            &$this->maSharedResourceWidgets[ +
                $this->GrassConfigResource->maszContents;
        $this->oGRASS =
            new grassOBJ($config);
        ...
    }
    function ParseURL() {

```

```

...
// initialize env.variable for GRASS
$this->oGRASS->set_mapset_grassrc();
$this->oGRASS->set_grass_env();
...
$cmd = "r.stats -acpln ".$this->layer;
$cmd .= " output=".$this->filename;
system($cmd)
...
}
}

```

Application example

In collaboration with Environmental Canada a Web-GIS with watershed analysis tools has been developed in order to dynamically extract basins and their land use statistical property (HYDRO1k, 2006). The data used for this project are the "North American HYDRO1k data", in particular the DEM, the flow accumulation and the flow direction maps (HYDRO1k, 2006), 1 Km resolution; the "North America Land Cover Characteristics" from USGS (NALCC, 2006), 1 Km resolution; the Canada river measurement stations table information (converted in shapefile) and the Canada province boundary shapefile. For this application 4 new widgets using grass functionalities have been implemented (see figure 1), they allow the user to: extract a basin selecting an existing river station as outlet (GrassWatershedPoint); extract a basin selecting a generic point as outlet (GrassWatershedPixel); derive and show statistical information of the land use classes in the extracted basin with a plot of an histogram of density (GrassWatershedStatistics); and clear the extracted basin (GrassWatershedClear).



Figure 1: Chameleon widgets using GRASS functionalities (from left to right: GrassWatershedPoint, GrassWatershedPixel, GrassWatershedStatistics, GrassWatershedClear).

Used in conjunction with the ROI (Region Of Interest) widgets they provide a watershed analysis tool bar; standard widget are used in order to derive a navigation tools, a legend and some other functionality (print, help and error report). In figure 2 you can see a screenshot of the application where all these tools are clearly visible.

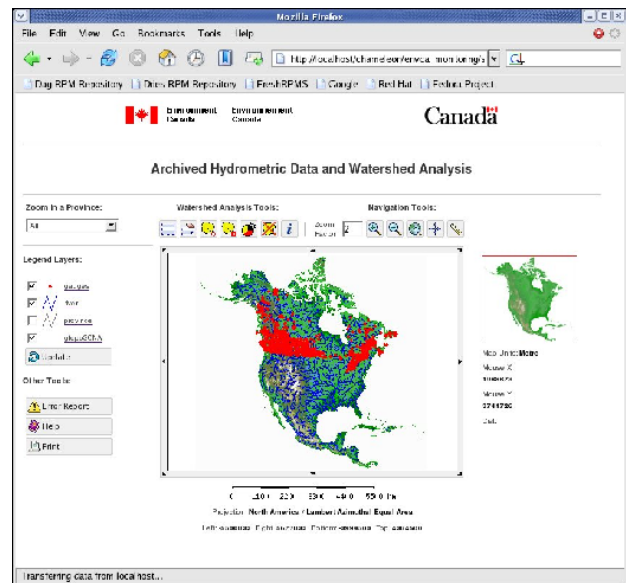


Figure 2: Screenshot of the example application.

As the rasters data to process is very large (73 million cells each), to prevent a too long response time of the server, as well as too big file generation, the GRASS principle of region has been used: a user in order to make analysis has to select a region with the ROI tools by drawing on the map a rectangle (see figure 3). *GrassWatershedPixel* or *GrassWatershedPoint* widgets will set this region (after reprojecting it in the current GRASS LOCATION projection) as the active analysis region in GRASS.

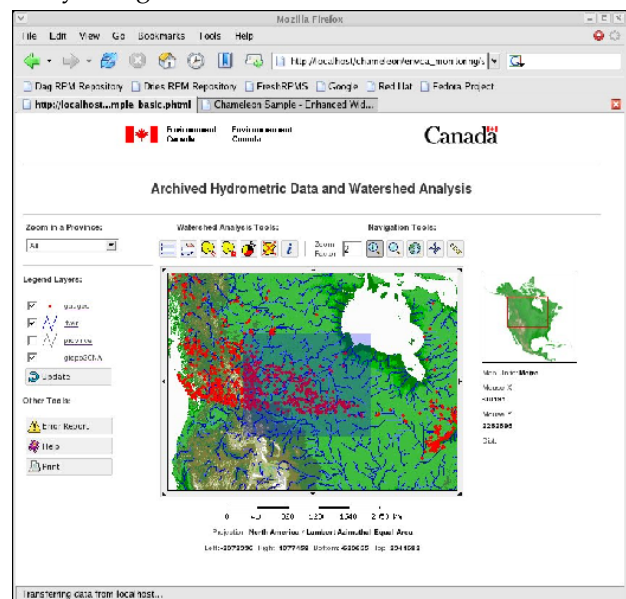


Figure 3: Analysis area selection by using the ROI tools.

In case of use of the *GrassWatershedPixel* or *GrassWatershedPoint* widgets without a selected region an error dialog box is popped-up and the calcu-

lation is not performed.

Once a region is selected it is possible to calculate a basin by selecting a station or by selecting a pixel on the map, figures 4 and 5 show the basins calculated with both these methods.

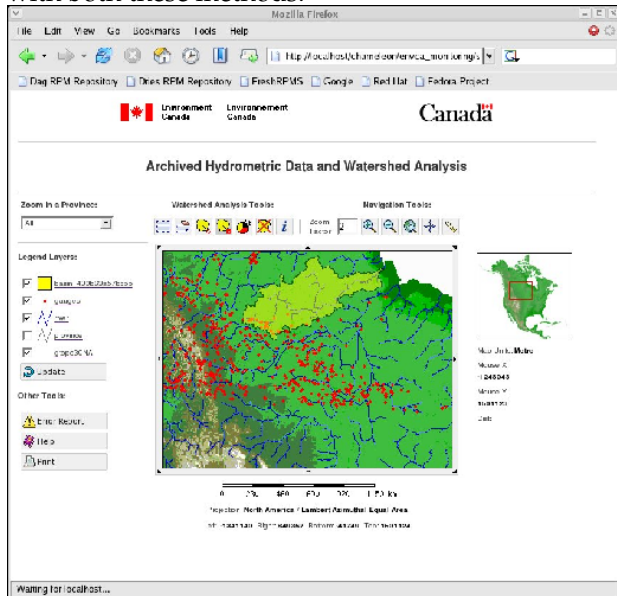


Figure 4: Basin calculated with GrassWatershed-Pixel.

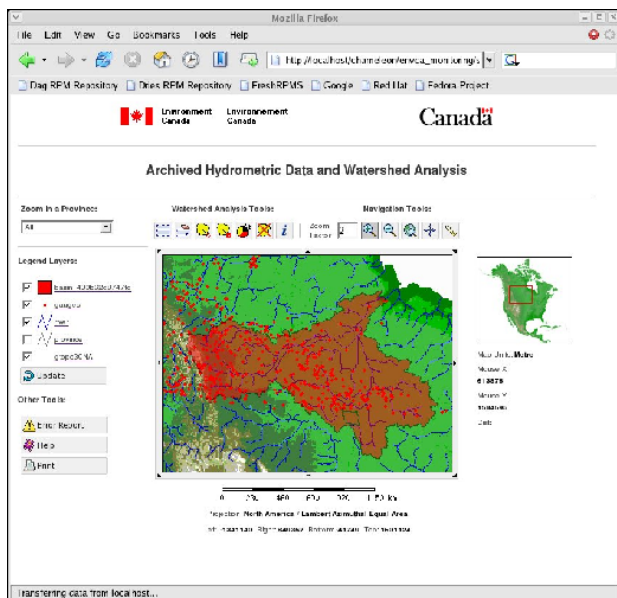


Figure 5: Basin calculated with GrassWatershed-Point.

In order to avoid too small basin calculation a new GRASS command has been developed (r.nearest.coord). Given a coordinate pairs, a map, a threshold and a search radius it return the coordinate of the nearest cell in the raster within the specified radius having a value bigger or equal than the threshold. Because of a river network can be

derived from a flow accumulation map by selecting the cells with a value higher than a threshold, if we use r.nearest.coord with such a map by using the the appropriate threshold value it will give us the coordinate of the nearest cell being on the river. This command, like the one for basin extraction (r.water.outlet), could result time consuming because of the high number of cells to handle: a key point in reducing calculation time is the use of an appropriate search radius parameter.

Once extracted a basin it is possible to calculate its statistical information based on another raster map by using the developed GrassWatershedStatistics widget: in the application a land use map is used. Once the widget is called a pop-up with three box is opened (see figure 6): the first box on the top shows the land use classes distribution (class number, class definition, area in square meters, number of cells and approximate percentage), the middle box shows some statistical parameters of the classes distribution (min, max, mean, standard deviation, etc.) and the bottom box shows a plot of the classes density (class vs. numerosity). All this values are related to the the current basin area calculated. Also in this case an error handling procedure is used and if this widget is clicked with no basin selected the boxes will show the warning message "WATERSHED NOT SELECTED !". The GrassWatershedClear widget simply allows a user to remove from the loaded map the basin calculated, as every new basin calculated has been load with a MapServer layer group value "GRASS": this command simply search and set the layer status property of these layers to MS_DELETE.

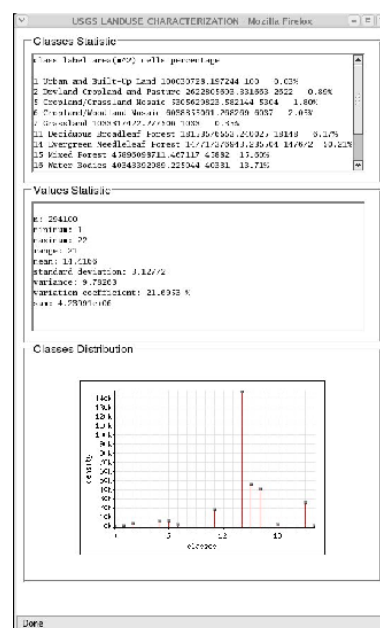


Figure 6: Watershed statistics calculated with the GrassWatershedStatistics widget.

Conclusions

A link between Chameleon and GRASS has been generated, showing how a Web-GIS with analysis functionalities can be easily developed in a fully free and open source environment that guarantee an accessible system, both in terms of coding and costing. Analysis procedures require time to be executed: considering that these tools has been developed for specific users (and not for the mass) the resulting waiting time for the application is acceptable, being in the order of a couples of seconds (to give an idea GrassWatershedPixel processing time to analyse a region of 1492 cols * 1006 rows is 8 seconds and for a region of 604 cols * 652 rows it takes 3 seconds). Because this solution has not been thought to provide a full GIS server service, but just to add specific GIS capabilities to the Web application, in order not to overload the server itself, a restricted users accessing procedure should be considered, as well as other technicals solution like the selection of a limited region for raster analysis used in the application example.

Acknowledgement

This work has been realized thanks to the funding of the "DRIN - Dottorati di Ricerca, Industria e Nuova impresa" project by Fondazione Politecnico di Milano. Special thanks to Prof. Maria Antonia Brovelli (Politecnico di Milano), to the DM Solution Group staff (Ottawa, Canada), and to Environment Canada for their support.

Bibliography

- DM Solution Group (2006) *Chameleon* <http://www.dmsolutions.ca/technology/chameleon.html>
- S. K. Jenson and J. O. Domingue (1988) Extracting Topographic Structure from Digital Elevation Data for Geographic Information Systems Analysis *Photogrammetric Engineering and Remote Sensing* 54 (11): 1593-1600.
- C. Ehlschlaeger (1989) Using the A* Search Algorithm to Develop Hydrologic Models from Digital Elevation Data *Proceedings of International Geographic Information Systems (IGIS) Symposium (Baltimore, MD, 18-19 March 1989)*: 275-281.
- R. Blazek and L. Nardelli (2004) The GRASS Server *Proceedings of the FOSS/GRASS Users Conference 2004 - Bangkok, Thailand, 12-14 September 2004* <http://gisws.media.osaka-cu.ac.jp/grass04/viewpaper.php?id=30>
- J. Y. Choi and B. A. Engel Real-Time Watershed Delineation System Using Web-GIS *Journal Computing in Civil Engineering* 17 (3):189-196
- University of Minnesota (2006) *Mapserver* <http://mapserver.gis.umn.edu>
- HYDRO1k Elevation Derivative Database (2006) *USGS* <http://edcdaac.usgs.gov/gtopo30/hydro/namerica.asp>
- NALCC, North America Land Cover Characteristics (2006) *USGS* http://edcsns17.cr.usgs.gov/glcc/na_int.html
- Hydro demo, watershed analysis demo site (2006) *IST* <http://w3.ist.supsi.ch:8001/geomatica/>
- Massimiliano Cannata
Institute of Earth Sciences (IST-SUPSI)
<http://w3.ist.supsi.ch:8001/geomatica/massimiliano.cannata> AT [supsi DOT ch](mailto:supsi@dot.ch)

Simultaneous simulation of hydrological and carbon cycle processes in a GIS framework

Integration of an existing distributed, process-oriented ecosystem model into GRASS GIS in combination with R

by Oliver Sonnentag

Introduction

The application of modeling techniques is a promising and widely used approach to many environmental problems and tasks in academia and industry, especially under circumstances in which direct mea-

surements are not feasible and also for prediction purposes. Process-oriented models simulate physical processes based on fundamental principles, often with some degree of empirical generalization.

The simultaneous simulation of carbon cycle and controlling hydrological processes using a *distributed*, process-oriented ecosystem model such as the Boreal Ecosystem Productivity Simulator (BEPS) developed by Liu et al. (1997, 1999, 2002) is very data-intensive. A variety of software including GIS, remote sensing image processing, and statistical packages has to be employed to pre-process the required input data sets

from different sources, as well as post-process, visualize, and analyze the model outputs.

BEPS has been developed in ANSIC as a stand-alone application on Microsoft's Windows[®] platform. The current version of BEPS contains a graphical user interface (GUI) based on the Microsoft Foundation Class[®] library written in Microsoft's Visual C++[®] (fig. 1). Previous versions of BEPS were executed from the Windows command line and through parameter files. The main purpose of the GUI is to provide a tool that simplifies model parameterization and allows on-screen display of spatial model outputs. As a result, user interaction is minimized which is beneficial, for example, for teaching purposes. However, the drawback of this increased user-friendliness is less flexibility with regard to future efforts that involve the further development and application of BEPS as a research tool.

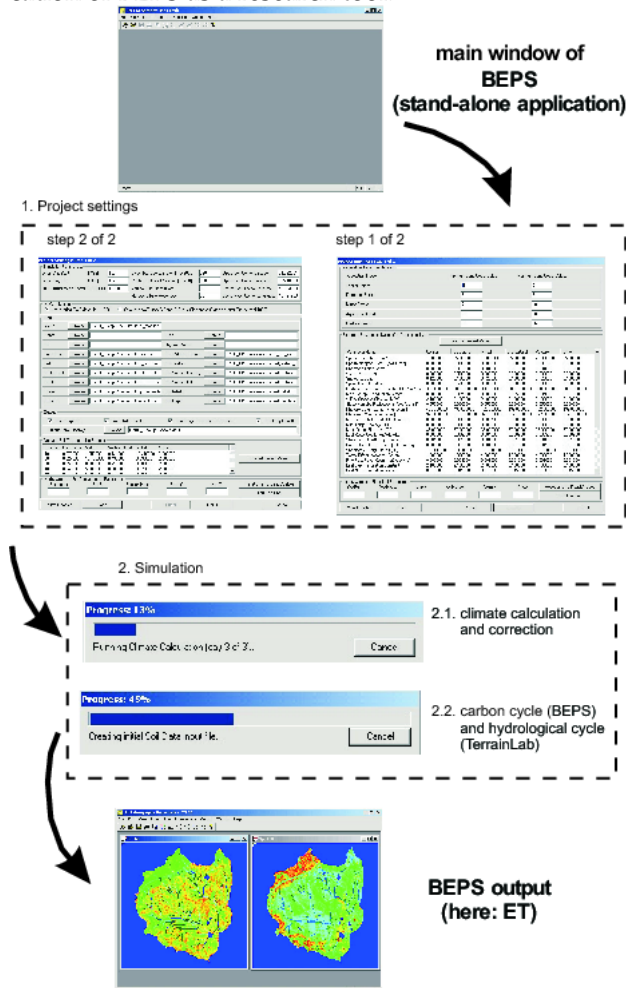


Figure 1: Application of the current version of BEPS containing a GUI on Microsoft's Windows platform.

My doctoral research involves the further development of BEPS towards a distributed, process-oriented ecosystem model of general applicability to individual peatlands of contrasting climatic, topo-

graphic, and (hydro) geologic settings in northern ecozones, able to reliably simulate the relevant hydrological and carbon cycle processes at the (sub-) watershed scale (1 - 100 km²).

In a first preparatory step I qualitatively evaluated the potential technical and scientific benefits of the application of BEPS in a GRASS GIS framework (v. 5.3.0) in combination with the mathematical and statistical programming language R (v. 1.9.1) on a Linux platform (SuSe[®] v. 9.0). A brief introduction to the main modeling hypotheses underlying BEPS, a description of the technical efforts undertaken for the integration of BEPS into GRASS GIS, and a discussion of the potential benefits of the integration are provided in the following sections.

The Boreal Ecosystem Productivity Simulator

BEPS was originally developed to assist in natural resource management and to simulate the carbon budget over the Canadian landmass. The most significant features of BEPS include modeling hypotheses related to the calculation of net primary productivity (NPP) and evapotranspiration (ET) in boreal and (sub-) arctic ecozones at a daily time-step.

NPP is the difference between gross primary productivity (GPP) and autotrophic respiration. In BEPS, GPP is calculated for overstory (Liu et al. , 1999) using a canopy-level photosynthesis model based on an instantaneous leaf-level photosynthesis model (Farquhar et al. , 1980). The upscaling of photosynthesis from leaf to canopy and from an instance of time to a day is accomplished with a spatial and temporal scaling scheme (Chen et al. , 1999). Autotrophic respiration for overstory is calculated as the sum of maintenance and growth respiration. Maintenance respiration is a product of biomass; growth respiration is calculated as a fraction of GPP (Liu et al. , 1999).

ET is calculated for several common overstory, understory, and soil components including snow sublimation from plants and soil. Transpiration from overstory and understory and evaporation from soil are calculated using the Penman-Monteith equation (Monteith , 1965). The upscaling of transpiration from leaf to canopy and from an instance of time to a day is accomplished by replacing stomatal resistance with canopy resistance using the same sunlit and shaded leaf separation approach as for NPP. Evaporation and sublimation from plants are both calculated from intercepted precipitation and the available energy for conversion of liquid or solid water to water vapour. Intercepted precipitation is assumed to be proportional to LAI, constrained by precipita-

tion, either rain or snow. Sublimation from snow is simply calculated based on available energy for conversion of solid water to water vapour. The soil water balance, and therefore the soil moisture content used for the calculation of the leaf water potential (Running et al. , 1988) required for the estimation of stomatal conductance (Jarvis , 1976), is calculated using a simple "bucket" model (Liu et al. , 1997), only considering vertical fluxes of water. In order to include topographic effects on the calculation of the soil water balance and the spatial variability of soil moisture content, the Distributed Hydrology Soil Vegetation Model (DHSVM) developed by Wigmosta et al. (1994) was adopted with several modifications and integrated in BEPS.

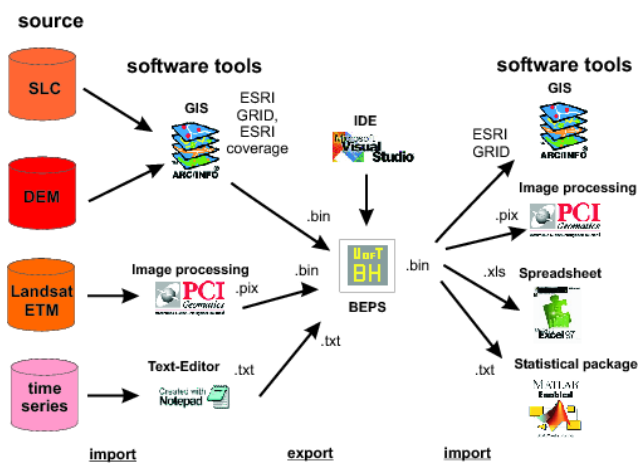


Figure 2: Workflow of the current version of BEPS as a stand-alone application on Microsoft's Windows platform.

BEPS operates with geographically prescribed, non-dynamic input data sets as square grid rasters and requires information on land cover type and leaf area index (LAI), both derived from satellite remote sensing images. Furthermore, spatially explicit input data of available soil water holding capacity, soil texture, initial soil water content, initial snow depth, and soil depth in addition to slope, aspect, and a watershed boundary derived from a digital elevation model (DEM) are required. Spatially explicit daily climatological input data sets (surface radiation, precipitation, and maximum, minimum, and dew point temperature) are constructed from measured time series data and corrected for temperature (slope and aspect) and surface radiation (elevation) before BEPS is run using the climate calculation and correction module.

Integration of BEPS and GRASS GIS in combination with R

The workflow for the application of the current version of BEPS on Microsoft's Windows platform generally encompasses several data conversions between various formats and applications (fig. 2)¹.

For example, in (Liu et al. , 1997, 1999, 2002) information on available soil water holding capacity was obtained from processing ESRI coverages from the Soil Landscapes of Canada (SLC) database² using ESRI's ArcInfo[®] workstation. Generally, satellite remote sensing images (e.g., Landsat TM/ETM+) used for land cover classification and LAI mapping are subjected to several corrections such as geometric, radiometric, and atmospheric, and require the application of a remote sensing image processing software package such as PCI Geomatica[®]. However, regardless of the software package used for the preparation of the respective data set, for the actual model run, all data sets have to be converted to binary files.

The binary model outputs obtained from BEPS can be displayed for a first visual inspection using the basic spatial data handling capabilities provided by the GUI of the current version (fig. 1). Further analysis and validation of the model outputs as well as subsequent map production requires the application of GIS, remote sensing image processing, and statistical packages (e.g., Matlab) (fig. 2).

In order to make use of the square grid raster, image processing, and graphic production capabilities of GRASS GIS in combination with the analytical capabilities of R (Bivand , 2000) for the modeling purposes with BEPS, the BEPS source code was "tightly coupled" (Brimicombe , 2003) with GRASS GIS. For this purpose, the two new commands

- r.climatecalc
- r.bepsterrain

for the climate calculation and correction module and the actual BEPS model, respectively, were created (fig. 3). After typing any of the two commands at the shell prompt, the user is asked to specify a file for certain simulation parameters such as day of the year for the start and the end of the simulation, corner coordinates of the square grid raster, and vertical and horizontal resolution, and the required spatially explicit input data sets, all of which are accessed as GRASS GIS raster files from the database in the specified mapset.

Over the years, the original version of BEPS has undergone continuous refinement and further development by different scientists. As a result, the current

¹This figure includes a word that is or is asserted to be a proprietary term or trade mark. Its inclusion does not imply it has acquired for legal purposes a non-proprietary or general significance, nor is any other judgment implied concerning its legal status.

²<http://sis.agr.gc.ca/cansis/nsdb/sic/intro.html>

implementation of BEPS' modeling hypotheses including utility functions such as data I/O consists of about 8500 lines of just sparsely documented and commented source code. A detailed analysis of the source code revealed that almost all data input/output (I/O) is accomplished through standard ANSI C library functions from the main() functions of the climate calculation and correction module and the actual simulation code, respectively. However, some of the input data sets used by BEPS during execution are accessed by the respective function from different modules when needed, which at the beginning lead to difficulties in tracing the data flow.

After centralizing all data I/O in the respective main() function, the programming effort for the integration of BEPS and GRASS GIS was minimal. The integration was accomplished by following the approach pursued by *r.example* which is available from the GRASS GIS CVS repository.

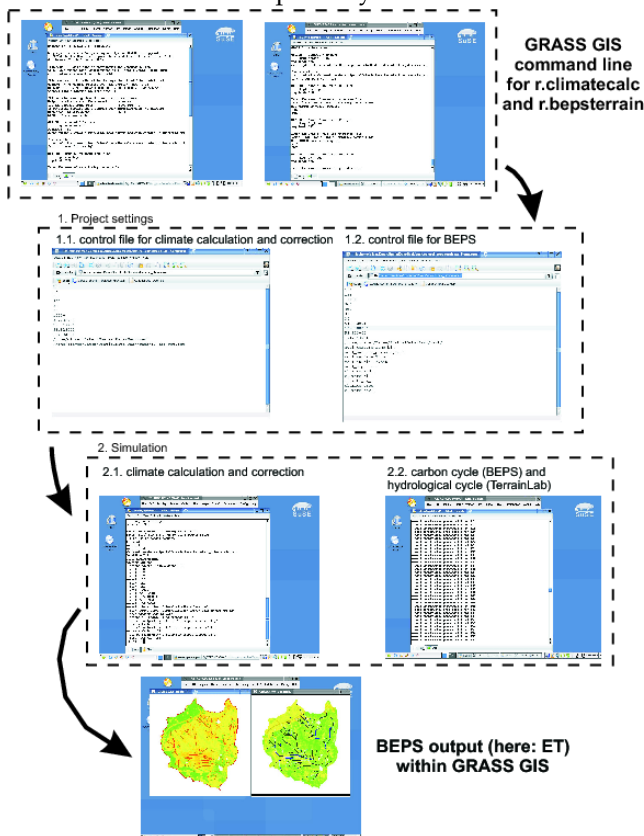


Figure 3: Application of BEPS integrated in GRASS GIS on a Linux platform.

Discussion and conclusions

The application of GRASS GIS as an integrated platform for the modeling purposes with BEPS in combination with R supersedes the application of multiple software packages and the implied conversion from proprietary data formats to binary files and back

(fig. 4). Furthermore, using Open Source software in the model development process allows the quick and straightforward adaptation of modeling source code as was demonstrated by simply adopting the well documented approach of *r.example*. Thereby, as demonstrated with the tight integration of BEPS and GRASS GIS, the full range of capabilities provided by GRASS GIS and R can be easily utilized by existing legacy modeling source code. As a result, GRASS GIS can be considered as the ideal computational platform for the modeling purposes with BEPS and their integrated use is of great technical and computational benefit. However, the current state-of-the-art for the integrated use of GIS and process-oriented models in general, regardless of the level of integration (loosely coupled, tightly coupled, and embedded after (Brimicombe, 2003)), is still purely a technical solution as to how both technologies "can share the same data rather than being integrated in terms of achieving compatible views of the world" (see Brimicombe, 2003, p. 170).

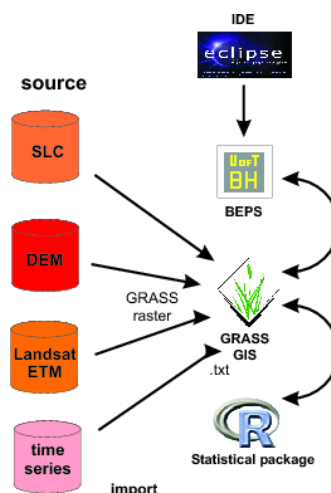


Figure 4: Workflow of BEPS integrated in GRASS GIS in combination with R on a Linux platform.

The reason for this deficiency lies in the fundamentally different abstraction of the world, represented by the underlying conceptual data models. Therefore, none of the currently applied ways of integrating GIS and process-oriented models has the capability to be considered as an enhanced means to provide additional insights into the functioning of complex environmental systems such as peatlands other than the insights gained from the application of the model itself.

As a result, for the purposes of my doctoral research, GRASS GIS is going to be of great importance from a technical and computational point of view, but is completely irrelevant for the achievement of my scientific goals.

Acknowledgements

The author would like to thank Dr. Tarmo Remmel for his helpful comments and discussions regarding this work.

Bibliography

- R. S. Bivand (2000) Using the R statistical data analysis language on GRASS 5.0 GIS database files. *Computers & Geosciences* 26: 1043-1052.
- A. Brimicombe (2003) *GIS, environmental modelling and engineering*. New York, USA: Taylor & Francis
- J. M. Chen, J. Liu, J. Chilar, M.L. Goulden (1999) Daily canopy photosynthesis model through temporal and spatial scaling for remote sensing applications. *Ecological Modelling* 124: 99-119.
- G. D. Farquhar, S. von Caemmerer, J.A. Berry (1980) A biochemical model of photosynthetic CO₂ assimilation in leaves of C₃ species. *Planta* 149: 78-90.
- P. G. Jarvis (1976) The interpretation of the variations in leaf water potential and stomatal conductance found in canopies in the field. *Philosophical Transactions Of The Royal Society Of London*: 593-610.
- J. Liu, J. M. Chen, J. Chilar, W. M. Park (1997) A process-based Boreal Ecosystem Productivity Simulator using remote sensing inputs. *Remote Sensing of Environment* 62: 158-175.
- J. Liu, J. M. Chen, J. Chilar, W. Chen (1999) Net primary productivity distribution in the BOREAS region from a process model using satellite and surface data. *Journal of Geophysical Research-Atmospheres* 104: 27735-27754.
- J. Liu, J. M. Chen, J. Chilar, W. Chen (2002) Net primary productivity mapped for Canada at 1-km resolution. *Global Ecology and Biogeography* 11: 115-129.
- J. L. Monteith (1965) Evaporation and environment. in *Proceedings of the 19th Symposium of the Society for Experimental Biology* New York, USA: Cambridge University Press.
- S. W. Running, J. C. Coughlan (1988) A general model of forest ecosystem processes for regional applications. 1. Hydrologic balance, canopy gas-exchange and primary production processes. *Ecological Modelling* 42: 125-154.
- M. S. Wigmosta, L. W. Vail, D. P. Lettenmaier (1994) A distributed hydrology-vegetation model for complex terrain. *Water Resources Research* 30: 1665-1679.

Oliver Sonnentag
 Department of Geography
 University of Toronto
 Toronto, ON, Canada
[oliver.sonnentag \(at\) gmail.com](mailto:oliver.sonnentag@gmail.com)

r.roughness – a new tool for morphometric analysis in GRASS

by Carlos Henrique Grohmann

Introduction

This article briefly describes `r.roughness`, a shell script written to calculate the surface roughness of raster surfaces. The method is based on Hobson (1972), where roughness is defined as the ratio between surface and plan area of square cells. This script will create square sub-regions with size defined by the user; in each sub-region, the real and planar areas will be calculated by `r.surf.area`, and the results (points at the centre of sub-regions) will be interpolated with `v.surf.rst`. The user also can set the tension and smooth parameters of interpolation.

Surface Roughness

Surface roughness (or topographical roughness) was first introduced as a morphometric parameter by

Stone and Dugundji (1965) and Hobson (1967, 1972). To Hobson (1972), one possible way to calculate it is the ratio between surface (real) area and flat (plan) area of square cells; in this approach, flat surfaces would present values close to 1, whilst in irregular ones the ratio shows a curvilinear relationship which asymptotically approaches infinity as the real areas increases.

Day (1979) describes surface roughness as the expression of non-systematic variability of the topographic surface, and used the dispersion of vector normals to surface plans as a roughness indicator to discriminate tropical karst stiles.

Ferrari et al. (1998) argue that surfaces with distinct characteristics can present the same roughness value, due the existence of interactions between the number and magnitude of terrain irregularities. Grohmann (2004), considers this method useful for morphological characterisation since it is mainly related with the shape of land-forms and not its elevation; thus, tectonically tilted areas have their expression shown, while it could be masked in a hypsometric map, as consequence of altimetric variations.

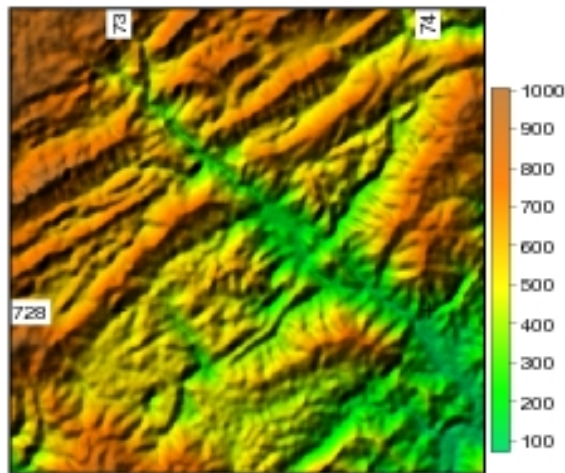


Figure 1: Study area.

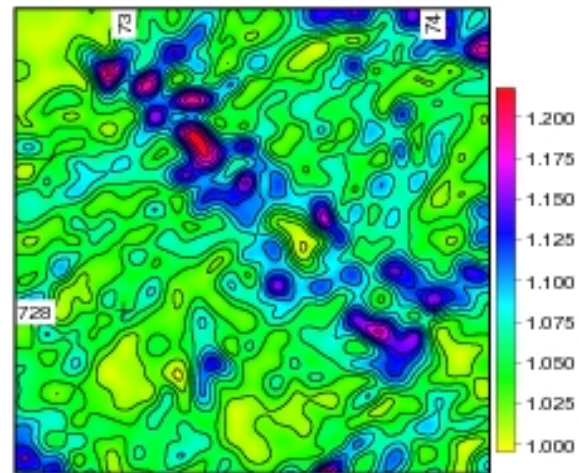


Figure 2: Surface roughness map, grid=500m.

The method has been applied in studies related to morphology of lake bottoms (Hakanson, 1974), as a discriminant of karstic areas (Day, 1979; Karmann et al., 1996; Ferrari et al., 1998), for structural compartmentation of sedimentary basins basement (Grohmann et al., 2005), in morphometric analysis of alkaline massifs (Roldan et al., 2006; Grohmann et al., 2007), in structural analysis of strike-slip shear zones (Steiner et al., 2006) and for macro-geomorphological compartmentation (Grohmann & Riccomini, 2006).

Usage and Examples

The script has five options: `map`, `grid`, `rough`, `tension` and `smooth`. `map` stands for the input raster surface and is the only required option. `grid` is the size of the sub-regions in which roughness will be calculated; the default value is 1000m. `rough` is the name for the output map; if a name is not provided, it will be set to `input_map_name.roughness.grid_size`. `tension` and `smooth` will be used by `v.surf.rst` for interpolation of roughness values; the default values are `tension=40` and `smooth=0.1`.

The examples presented are from an area located in southeastern Brazil, southern region of São Paulo State. Local geology consists of NE-SW trending metapelitic and metacalcareous rocks where karstic landscapes developed over the carbonatic rocks, with altimetric differences up to 700m between non-carbonatic (pelitic, psamitic and granitic) crests and karstic valley bottoms. NW-SE trending dikes cut across the area and have a strong influence on geomorphological development (Fig. 1). Surface roughness was calculated for grid sizes of 500, 1000 and

2000m.

With a grid size of 500m (Fig. 2), a good correlation with land-forms can be seen. Higher roughness values are related with the Bethary River valley, developed over a NW-SE trending dike. Also, karstic valleys have smaller roughness values than non-carbonatic crests. The general picture of the features present in (Fig. 2) can be seen in the map for grid size of 1000m (Fig. 3), although is not possible to individualise the answer from each carbonatic unit. A grid of 2000m (Fig. 4) does not give much information, indicating that land-forms within this area cannot be well described with a wavelength this large.

Concluding Remarks

Surface roughness is a useful parameter for morphological compartmentation. `r.roughness` is a shell script that automises the process, but users must be aware that it uses `r.surf.area` to calculate both real and planar area for each grid cell (sub-regions) and that raster resolution plays an important role on area estimations.

The script is available through GRASS Wiki site ³ in two versions: `r.roughness` for GRASS 6.1+ and `r.roughness60` for GRASS 6.0.x.

Acknowledgements

This work was supported by FAPESP grant 04/06260-5 (C.H.G. Doctor's Degree Fellowship). The author is thankful to Tomás Senabre (Spain) for his feedback which helped to improve the script usability.

³http://grass.gdf-hannover.de/wiki/GRASS_AddOns

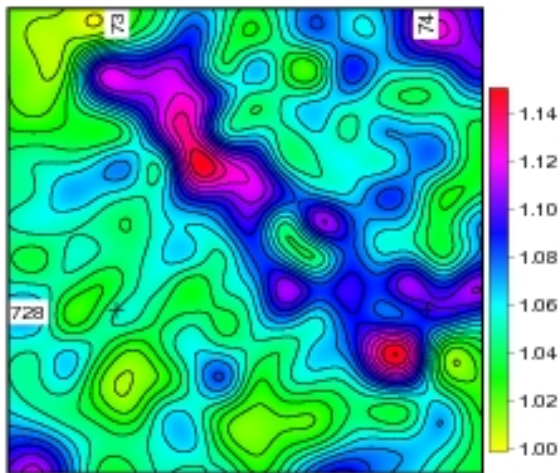


Figure 3: Surface roughness map, grid=1000m.

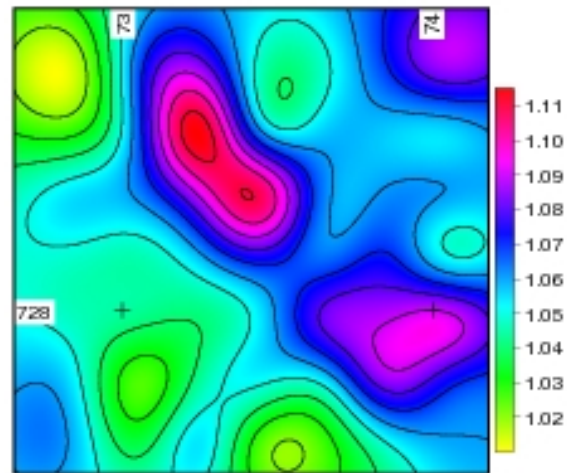


Figure 4: Surface roughness map, grid=2000m.



Bibliography

- M. J. Day (1979) Surface roughness as a discriminator of tropical karst styles *Zeitschrift für Geomorphologie*, Suppl.-Bd.32:1-8.
- J. A. Ferrari, S. T. Hiruma, I. Karmann (1998) Caracterização morfométrica de uma superfície cárstica do Vale do Ribeira, São Paulo (Núcleo Caboclos - PETAR) *Revista do Instituto Geológico*, 19:9-17.
- C. H. Grohmann (2004) Morphometric analysis in Geographic Information Systems: applications of free software GRASS and R *Computers & Geosciences*, 30:1055-1067. <http://dx.doi.org/10.1016/j.cageo.2004.08.002>
- C. H. Grohmann, C. Riccomini, F. Paula e Silva and H. K. Chang (2005) Compartimentação tectônica da Bacia Bauru no centro-oeste do Estado de São Paulo *In: X Simpósio Nacional de Estudos Tectônicos / IV International Symposium on Tectonics, Curitiba-PR*, Boletim de Resumos Expandidos:25-27.
- C. H. Grohmann and C. Riccomini (2006) Análise regional de rugosidade de relevo *In: XLIII Congresso Brasileiro de Geologia, Aracaju-SE* Anais:97.
- C. H. Grohmann, C. Riccomini and F. M. Alves (2007) SRTM-based morphotectonic analysis of the Poços de Caldas Alkaline Massif, southeastern Brazil *Computers & Geosciences (in press)*. <http://dx.doi.org/10.1016/j.cageo.2006.05.002>
- L. Hakanson (1974) A mathematical model for establishing numerical values of topographical roughness for lake bottoms *Geografiska Annaler. Series A, Physical Geography*, 56:183-200.
- R. D. Hobson (1967) FORTRAN IV programs to determine the surface roughness in topography for the CDC 3400 computer *Computer Contribution 14, State Geol. Survey Kansas*, 28p.
- R. D. Hobson (1972) Surface roughness in topography: quantitative approach *In: Chorley, R.J., 1972. Spatial analysis in geomorphology*, 225-245.
- I. Karmann, R. F. Pereira and J. A. Ferrari (1996) Índice de rugosidade: parâmetro morfométrico da intensidade de relevo. Exemplo do carste da bacia do Rio Una, Bahia *In: XXXIX Congresso Brasileiro de Geologia, Salvador*, Anais:575-579.
- L. F. Roldan, S. S. Steiner, C. H. Grohmann and R. Machado (2006) Análise morfométrica da região do Domo de Lages, SC *In: XLIII Congresso Brasileiro de Geologia, Aracaju-SE* Anais:266.
- S. S. Steiner, R. Machado and C. H. Grohmann, C. H. (2006) Análise morfométrica da estrutura divergente na região do Rio Paraíba do Sul, Rio de Janeiro *In: XLIII Congresso Brasileiro de Geologia, Aracaju-SE* Anais:97.
- R. O. Stone and J. Dugundji (1965) A study of microrelief – its mapping, classification and quantification by means of a Fourier analysis *Engineering Geology*, 1:89-187.

Carlos Henrique Grohmann
Institute of Geosciences
University of São Paulo, Brazil
<http://www.igc.usp.br>
guano AT usp br

Resampling SRTM 03"-data with kriging

by Carlos Henrique Grohmann

Introduction

SRTM is now been widely used as source for DEMs. The data is distributed at horizontal resolution of 30 meters (aprox. 1arcsec) for areas within the U.S.A. and at 90 meters resolution (aprox. 3arcsec) for the rest of the world.

A resolution of 90m can be considered suitable for small or medium-scale analysis, but it is too coarse for more detailed purposes. The present alternative is to interpolate the DEM at a finer resolution. It won't increase the level of detail of the original DEM, but it will lead to a surface where there is coherence of angular properties (i.e., slope, aspect) between neighbouring pixels (Valeriano et al., 2006), an important characteristic when dealing with terrain analysis.

The purpose of this article is to present the steps necessary to improve the resolution of a DEM using variogram modelling and kriging, as well as a brief comparison of the results with those obtained with interpolation by Regularised Splines with Tension (RST).

GRASS users must be aware of the half-pixel shift in SRTM "finished" data from USGS web site ⁴ and use proper tools to import SRTM data, as pointed by Neteler (2005).

Geology and geomorphology of the study area

The area used as example is located in southeastern Brazil, southern region of São Paulo State (Fig. 1). In general terms, local geology consists of NE-SW trending metapelitic and metacalcareous rocks (Fig. 2) of Precambrian age, deformed by the Brasiliano/Panafrican Orogenic Cycle (600-450 Ma) (Campanha and Sadowski, 1999) and affected by Cretaceous brittle tectonics and basic dike emplacement.

Karstic landscapes developed over the carbonatic rocks, with altimetric differences up to 700m between non-carbonatic (pelitic, psamitic and granitic) crests and karstic valley bottoms; the structural pattern of the area, alternating elongated ranges of non-carbonatic rocks and lowered karstic zones, gives origin to mixed recharge systems, with important allogenic water input (Karmann and Ferrari, 2000).

⁴<http://seamless.usgs.gov>

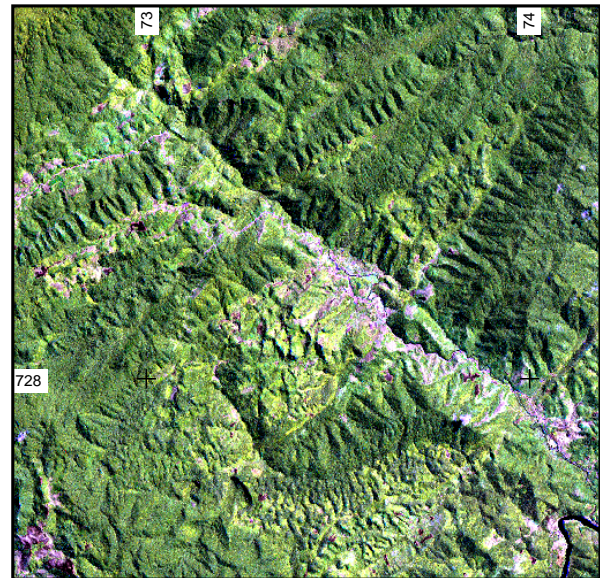


Figure 1: Landsat 7 ETM+ image of the study area. UTM Coordinate System, Zone 22, Southern Hemisphere

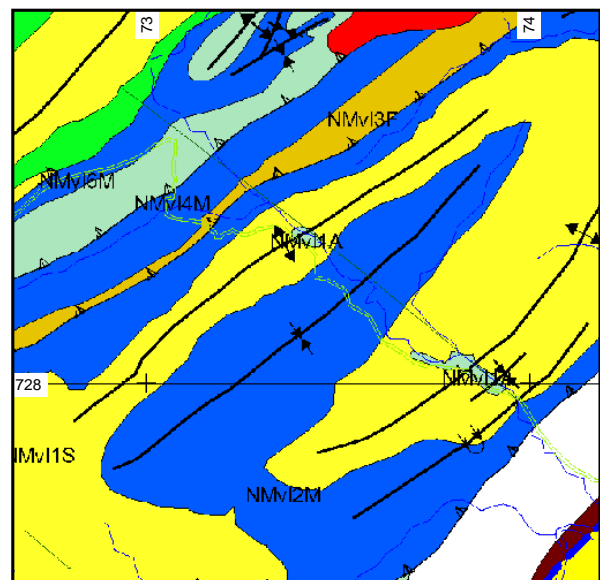


Figure 2: Simplified geology of study area (Campanha, 2003).

Variogram modelling

In this section we will deal with variogram modelling and kriging of a GRASS raster within the R statistical language environment. If you are not familiar with the

R interface, please refer to [Bivand \(2005\)](#). Kriging procedure is partially based on GRASS-WIKI⁵ and on the Short draft introduction to R/GRASS interface⁶.

The variogram is a tool that allows to describe quantitatively the variation, in space, of a regional phenomenon. Elevation data are usually expected to be high spatially dependent (high similarity of data at short distances); the noise present in such data, that is, low similarity of data at distances close to the grid size, can be evaluated as the rate of nugget effect in variograms ([Valeriano et al., 2006](#)). Data with smooth variations (such as water levels and land-forms) often have variograms with a region of low slope near the zero distance, that can be best modelled by a Gaussian model ([Burrough, 1987](#)).

According to [Valeriano \(2002\)](#), variograms calculated with linear trend residues of topographic data have adequate fits to classical variogram models, which present a clear and defined sill. Residues of trend surface analysis are used to guarantee geostationarity of data being modelled.

Since DEMs datasets can be very large, variogram calculation and kriging interpolation can become a very time-consuming tasks. The basic idea is to choose a representative subset of the study area, and calculate the variogram over this subset. This variogram model will then be used by kriging to interpolate the whole dataset at a finer resolution.

In this example, I worked with three regions: karst (the original dataset at 90m resolution), karstsub (a subset, 90m resolution), and karst30 (same extents as karst, 30m resolution).

```
>g.region n=7289760 s=7274730 w=726480 e=741870
res=90 save=karst
>g.region n=7284000 s=7280130 w=731630 e=735860
res=90 save=karstsub
>g.region n=7289760 s=7274730 w=726480 e=741870
res=30 save=karst30
```

We start working with the karstsub region. Start the R session, call needed libraries, set up grid parameters and import the srtm raster file.

```
>R
>system("g.region region=karstsub");
>library(spgrass6);library(spatial);library(gstat);
>G <- gmeta6();
>grd <- GridTopology(cellcentre.offset=
c(G$west+(G$ewres/2), G$south+(G$nsres/2)),
cellsize=c(G$ewres, G$nsres),
cells.dim=c(G$cols, G$rows));
>mask_SG <- SpatialGridDataFrame(grd,
data=list(k=rep(1, G$cols*G$rows)),
proj4string=CRS(G$proj4));
>srtm <- readFLOAT6sp("srtm_v2");
> names(srtm)
```

```
[1] "srtm_v2"
```

Now we can calculate a variogram and see how it looks (Fig. 3).

```
>variog1<-variogram(srtm\$\srtm_v2~coords[,1]+
coords[,2],loc=srtm, srtm);
>plot(variog1);
```

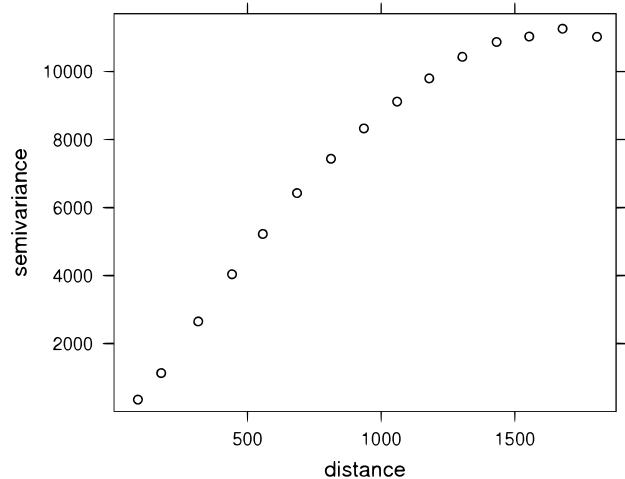


Figure 3: Calculated variogram

We can fit a first model "by eye"(Fig. 4).

```
>vrg. eye<-(vgm(psill=11000,model="Gau",range=800,
nugget=50));
>plot(variog1, model=vrg. eye);
```

We can check if our eyes are good and ask gstat to calculate the model parameters.

```
>vrg. fit<-fit.variogram(variog1,vrg. eye);
>vrg. fit;
```

model	psill	range
1 Nug	378.0942	0.0000
2 Gau	9982.4331	673.9686

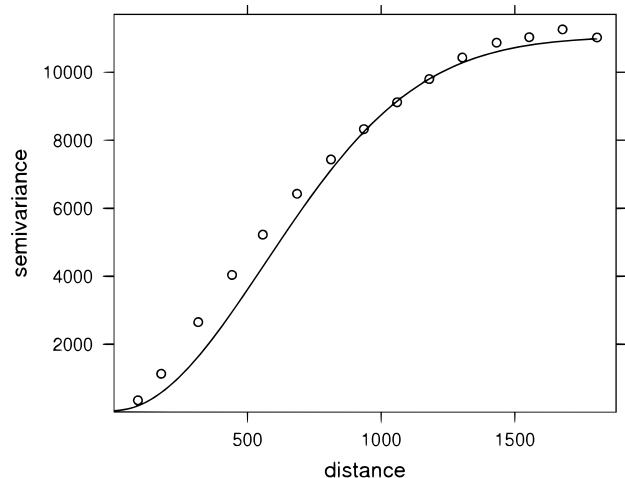


Figure 4: Calculated variogram and adjusted Gaussian model

⁵http://grass.gdf-hannover.de/wiki/How_to_interpolate_point_value_using_kriging_method_with_R_and_GRASS_6

⁶http://www.geog.uni-hannover.de/grass/statsgrass/grass_geostats.html

If we want, we can adjust the parameters according to those given by `gstat`. Since the initial part of the curve of the Gaussian model and the nugget effect will both have the effect of smoothing the interpolated surface, we can concentrate on adjust just this part of the model (Fig. 5). Using a nugget effect value of zero would result in a very noisy surface, while a large value produces a smooth surface, maybe at the cost of loosing detail.

```
>vrg.eye2<-(vgm(psill=9000, model="Gau", range=550,
nugget=5));
>plot(variog1, model=vrg.eye2);
```

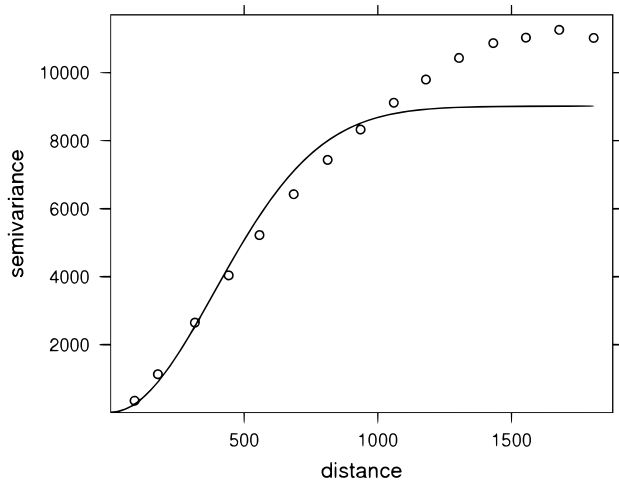


Figure 5: Gaussian model adjusted to the initial part of the curve

After the variogram model is adjusted, it is time to interpolate the new DEM. First, we need to set the region back to the full area of the DEM, but with finer resolution, re-set (in this case, overwrite) the `gmeta6()` parameters and import the `srtm_v2` raster. Then, we need to add a very small random variation into the coordinates of the points, to avoid interpolation artifacts later. We add the noise with `jitter`, and create a new object of class `SpatialPointsDataFrame`.

```
>system("g.region region=karst30");
>G <- gmeta6();
>grd <- GridTopology(cellcentre.offset=
c(G$west+(G$ewres/2), G$south+(G$nsres/2)),
cellsize=c(G$ewres, G$nsres),
cells.dim=c(G$cols, G$rows));
>mask_SG <- SpatialGridDataFrame(grd,
data=list(k=rep(1, G$cols*G$rows)),
proj4string=CRS(G$proj4));
>srtm <- readFLOAT6sp("srtm_v2");

>coords<-coordinates(srtm);
>jcoords <- cbind(jitter(coords[,1]),
jitter(coords[,2]));
>cat<-as.data.frame(srtm$srtm_v2);
>srtm2<-SpatialPointsDataFrame(jcoords,cat,
proj4string=CRS(G$proj4));
```

Now we can (finally) use kriging to interpolate the new surface. The `maxdist` parameter specifies that only points within this distance are used for interpolation. We can visualise it with `image` (Fig. 6).

```
>OK_pred <- krige(srtm$scat~1, loc=srtm2,
newdata=mask_SG, model=vrg.eye2, maxdist=210);
>names(OK_pred);
>image(OK_pred,"var1.pred");
>image(OK_pred,"var1.pred",xlim=c(731630,735860),
ylim=c(7280130,7284000), col=topo.colors(80))
```

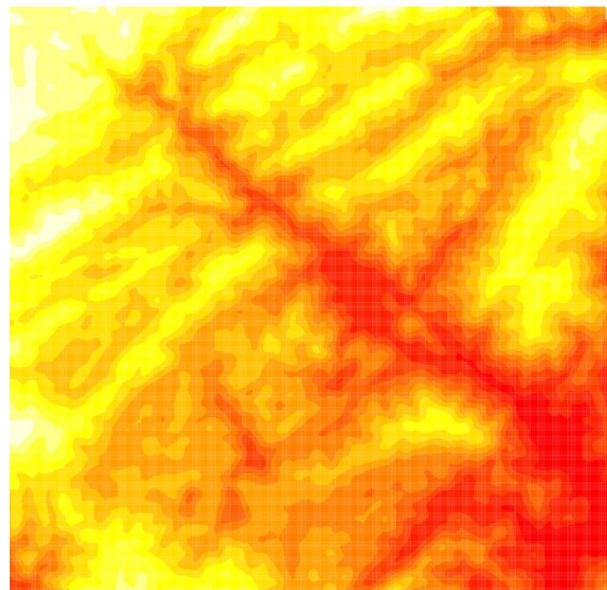


Figure 6: Interpolated surface, image display

The `levelplot` function of the `lattice` library lets you control the aspect ratio and draws a legend by default (Fig. 7).

```
>library(lattice);
>levelplot(OK_pred$var1.pred~OK_pred@coords[,1]+
OK_pred@coords[,2], OK_pred, aspect = "iso",
main = "ordinary kriging predictions", xlab="",
ylab="",scales = list(y = list(rot = 90)),
col.regions=topo.colors(80))
```

Let's send the raster back to GRASS, and finish our R session for now.

```
>writeRast6sp(OK_pred,"karst.krig",
zcol="var1.pred");
>quit()
```

Results and Discussion

Interpolation by RST has been successfully used for void filling of SRTM data, and can also be used to interpolate the DEM at a finer resolution. For more on this subject, please refer to [Neteler \(2005\)](#).

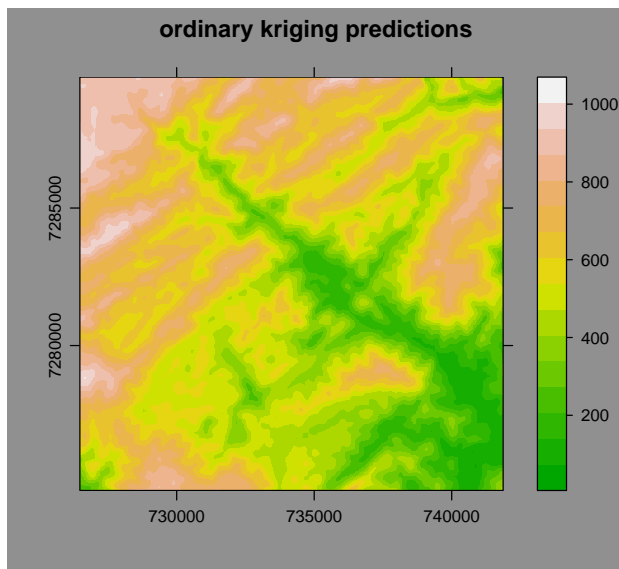


Figure 7: Interpolated surface, levelplot display

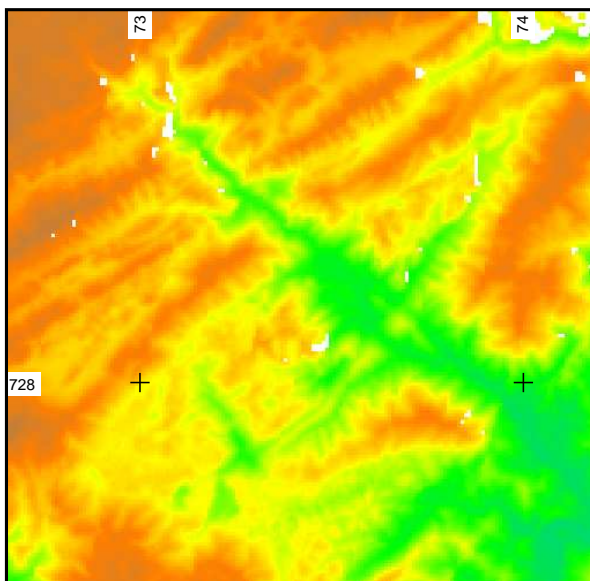


Figure 8: Original SRTM V2 data

As a means of comparison with the results obtained with kriging, the SRTM dataset was also interpolated at 30m resolution using RST (with default values). Figure 8 is the original SRTM data, with intrinsic noise, artifacts and voids; figures 10 and 11 are shaded relief maps for the interpolated DEMs with RST and kriging, respectively.

From the figures above, we can see that RST map has a sharper look and does a better job about void filling (note the dark areas in the upper right corner of kriging map), but linear artifacts are still present. The krigged map has a smoother appearance, with crests not so well defined as in the RST map, but the linear artifacts were essentially removed.

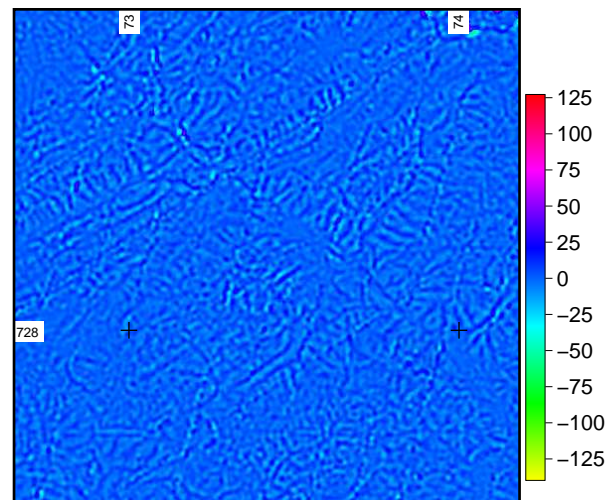


Figure 9: Difference map of krigged and RST values

In order to compare both interpolations, the difference between them was calculated with `r.mapcalc` as `krigged_map minus RST_map`. In Figure 9, positive values (areas where kriging values are higher than RST ones) have colors ranging from blue to red and negative values have colors from cyan to yellow; The greatest differences are in valley bottoms and in crests due the smoothing behaviour of the kriging function.

To better evaluate the differences of interpolated products, standard deviation (SD) maps were calculated with a 7x7 moving window (using `r.neighbours`). From Figure 12, we see that the SD of kriging interpolation is a little smaller than the SD of RST. If a difference map is made from the SD map (as `kriging_SD minus RST_SD`, Figure 13), the presence of artifacts in RST interpolation is enhanced; the extreme values are related to areas of voids in the original data, that were not completely filled by kriging.

Table 10 presents a statistical summary of the interpolated DEMs and their derivatives slope, aspect and standard deviation (all derivatives were calculated with a 7x7 window).

The purpose of this article was to present a simple guide on how to increase SRTM data resolution using kriging within the R environment. Also, a brief comparison with results obtained by RST was made.

Conclusions

RST interpolation is a strong method, suitable for void filling, and produces good results, although it does not eliminate the artifacts inherent to SRTM data. A more detailed study concerning the fine-tuning of RST parameters may point interesting directions on the use of this tool.

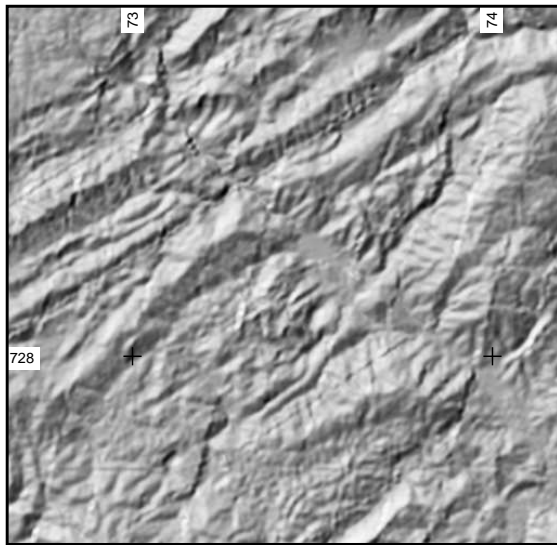


Figure 10: Shaded Relief of RST interpolated DEM, illuminant at 315°, 30° above the horizon

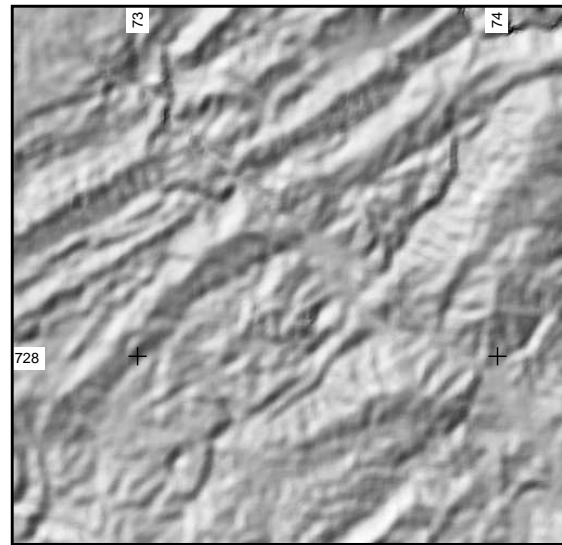


Figure 11: Shaded Relief of kriging interpolated DEM, illuminant at 315°, 30° above the horizon

Kriging interpolation may be a more laborious task, since it involves variogram modelling prior to interpolation. Care must be taken on all steps of the process. The use of the `maxdist` option allows the user to perform a good adjust of the variogram model just on the initial part of the curve, but larger voids will become anomaly areas or will remain unfilled. Nugget effect will act as a smoothing factor; a small value is sufficient to eliminate noise, while a larger one may obliterate terrain features.

In case of large voids in SRTM data, one possible approach is to first fill the voids with RST (`r.fillnulls`) and then resample with kriging.

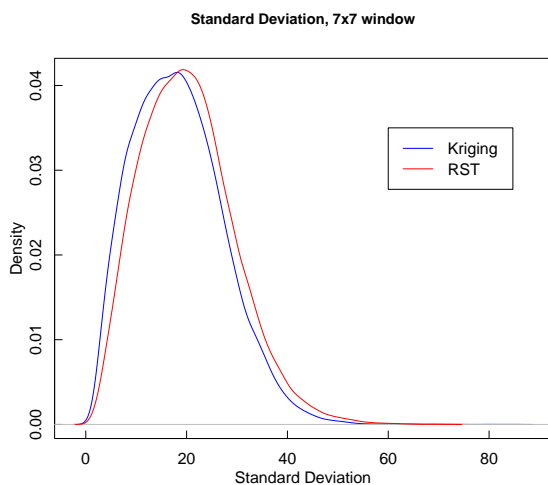


Figure 12: Density functions for standard deviation

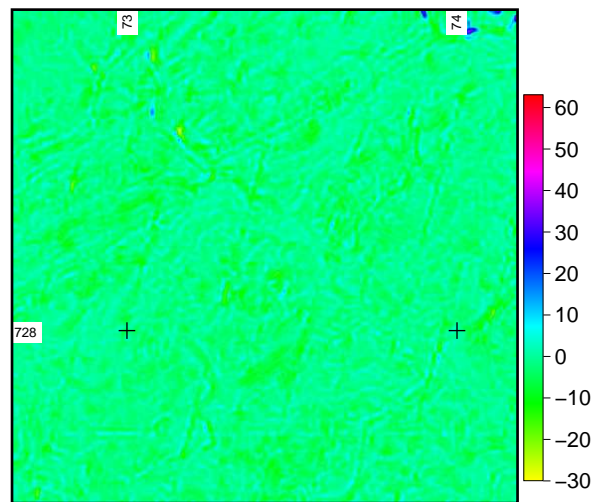


Figure 13: Difference map for standard deviations

Acknowledgements

This work was supported by FAPESP grant 04/06260-5 (C.H.G. Doctor's Degree Fellowship). The author is thankful to Roger Bivand, Márcio Valeriano, Markus Neteler, Hamish Bowman and Dylan Beaudette for discussions, ideas and support on the subject.

Bibliography

R. Bivand (2005) Interfacing GRASS 6 and R. *GRASS-News*, 3:11-15.

	krig_DEM	krig_slope	krig_aspect	krig_stdev	RST_DEM	RST_slope	RST_aspect	RST_stdev
Min	71.37	0	-180	0.466	66.55	0	-179.999	0.9144
1st Qu.	430.87	9.491	-112.845	11.545	430.42	10.29	-111.784	13.4169
Median	566.56	15.667	-5.742	17.734	567.06	16.74	-5.602	19.6106
Mean	551.38	15.713	-11.547	18.39	551.38	16.69	-11.034	20.2738
3rd Qu.	690.73	21.495	77.662	24.189	690.65	22.61	78.142	26.0236
Max	1005.2	54.47	179.997	85.725	1005.6	49.58	179.995	71.6148

Table 1: Statistical summary of interpolation products.



- P. A. Burrough (1987) Spatial aspects of ecological data. In: Jongman, R.H.; ter Braak, C.J.F.; Tongeren, O.F.R. (Eds), *Data analysis in community and landscape ecology*. Pudoc, Wageningen, pp.213-251.
- G. A. C. Campanha and G. R. Sadowski (1999) Tectonics of the southern portion of the Ribeira Belt (Apiaí Domain) *Precambrian Research*, 98:31-51.
- G. A. C. Campanha (2003) *Mapa Geológico da Folha Itararé (SG-22-X-B) em escala 1:250.000*.
http://www.igc.usp.br/pessoais/ginaldo/itarare_ld/Itarare2003.zip
- I. Karmann and J. A. Ferrari (2000) Carste e cavernas do Parque Estadual Turístico do Alto Ribeira (PETAR), sul do Estado de São Paulo. In: Schobbenhaus, C.; Campos, D.A.; Queiroz, E.T.; Winge, M.; Berbert-Born, M. (Eds.) *Sítios Geológicos e Paleontológicos do Brasil*.
<http://www.unb.br/ig/sigep/sitio043/sitio043.htm>
- M. Neteler (2005) SRTM and VMAP0 data in OGR and GRASS *GRASS-News*, 3:2-6.

E. J. Pebesma (2004) Multivariable geostatistics in S: the gstat package. *Computers and Geosciences*, 30: 683-691.

M. M. Valeriano (2002) Modelos digitais de elevação de microbacias elaborados com krigagem. *Information and Documentation Service (SID)*, INPE, Technical Report INPE-9364-RPQ/736, 54pp.

M. M. Valeriano, T. M. Kuplich, M. Storino, B. D. Amaral, J. N. Mendes Jr. and D. J., Lima (2006) Modeling small watersheds in Brazilian Amazonia with shuttle radar topographic mission-90m data. *Computers and Geosciences*, in press, available on-line.
<http://dx.doi.org/10.1016/j.cageo.2005.10.019>

Carlos Henrique Grohmann
 Institute of Geosciences
 University of São Paulo, Brazil
<http://www.igc.usp.br>
 guano AT usp br

Interview with Michael Barton

Michael Barton is geoarchaeologist by training. His research interests are on long-term human ecology and human-environmental interaction at regional scales. He is professor & curator of archaeology/ethnology at the Arizona State University, Tempe, USA

Welcome to the first interview series in 2006. Could you start by telling us a bit about yourself, what is your profession, where do you live, which OS, GRASS version are you using etc? How did you

first get into contact with GRASS?

Macintosh is my preferred computing platform. When the Mac OS switched to BSD Unix, I was interested in trying GRASS which I had heard of for many years. My colleagues and collaborators at the University of Valencia were also interested in GRASS at that time and beginning to work with Linux. By 2001, I was increasingly frustrated with ArcView, both the functioning of the program and the lack of Mac support. In fact, I wrote a letter to ESRI customer support about the lack of Mac support in particular and lack of support in general which sur-

prised me given that I am at a major US university which has had a campus-wide ESRI license since the late 1990s. I even said that without support for Mac/Unix platforms, I was strongly considering switching all my research and teaching to GRASS. ESRI's response was essentially, we are going to follow the commercial market. So I followed through on the threat and here I am.



Figure 1: Michael Barton

Which GIS/RS software did you use first? Any commercial ones, did you drop them in favour of GRASS? And if yes, why?

In my first year of graduate school still at the University of Kansas where I did my undergrad work was working as an RA (Research Assistant - a graduate student position to help a professor with a research project) for an archaeologist, doing spatial analysis of a late glacial archaeological site in Bosnia. We had thousands of artifacts from the excavation plotted in 3D and I had to do repeated back plots and plan views of artifacts. I could see in my mind the 3D artifact distribution, but re-plotting every artifact on graph paper for many views was too laborious to do. I heard that the Kansas Geological Survey was working on a computer mapping program. So I went to talk with the programmers to ask them if it was possible for a computer to take our xyz coordinates and make a contour map of artifact densities. They said not yet, but we're working on it. This was in 1976. I was looking for that kind of program from that point on. KGS (Kansas Geological Survey) eventually came out with *Surface* and *Surface II* that did this though it was far too late to help me in the RA

job.

When I did my PhD thesis in the mid-1980's, I used a version of Atlas Graphics to make one of my maps. It seemed like a good idea, but was very difficult to use. I didn't have access to the Unix workstations that *ArcInfo* and *GRASS* ran on in those days. After I came to ASU in 1987, I started my current long-running field project in eastern Spain. When I was planning for the first field season, I knew I wanted to do digital mapping. I first bought a CAD program with my grant money. We mapped survey areas and traced contour lines. Then *MapInfo* for Windows came out in 1990. I knew that GIS was what I REALLY needed, and moved all the first season's data to that.

MapInfo was my first real experience with GIS. My experience got a number of people interested in GIS here at ASU and some of them still use *MapInfo*. It is a very nice program in many respects. It is very sensible from the user point of view and quite powerful. Its main drawback is its poor support for raster spatial data. As I moved from making maps to spatial analysis, this became an increasing drawback for my research program. I remember following the instructions in Ian Johnson's *MapInfo for archaeology* book on how to do a Local Density Analysis with *MapInfo*. Eventually, I made the switch to *ArcView* 3 in the late 1990's. *ArcView* was pretty worthless before this. But with version 3 and some effort customising the program it was pretty functional. There were still a number of things that were much easier in *MapInfo* (especially database manipulation), but the combination of raster and vector overrode these annoyances.

As I said earlier, by 2000 and 2001, however, I had become increasingly frustrated with *ArcView* especially the bugginess and lack of Mac support. By 2001 (when I taught my first spatial technologies course here at ASU) I was running *ArcView* under Virtual PC and it was regularly crashing (it did the same for PC users). This was when I began seriously looking into *GRASS*. I had a sabbatical in Valencia in the fall of 2003, that included teaching an informal GIS seminar. This gave me the opportunity to really look into *GRASS*. I started to become proficient in the program that fall. The following spring (2004) I taught spatial technologies again here at ASU and used *GRASS* along with *ArcView* as a recommended program. This forced me to learn it much better (you HAVE to learn it well to teach it). Along with my colleagues in Valencia, I moved all of my data to *GRASS* that year. Now I'm using it as the primary GIS and spatial modelling tool for a large, 5-year research project in the Mediterranean, funded by the US National Science Foundation. I taught *GRASS* GIS to archaeology doctoral students at Valencia again last summer and am using it in my

remote sensing course at ASU this spring.

Do you recommend people (students, colleagues) to use GRASS instead of other software, and if yes, why?

I highly recommend it for 3 reasons:

1. It is simply an excellent piece of geospatial analysis and visualisation software. It is as or more powerful than any other GIS I've worked with and much easier to accomplish complex spatial analysis tasks in than other software.
2. Because it is open source, it can be modified to suit a given research project. The development team is highly responsive. I never could get a response from ESRI even though I've used it in a number of high-visibility projects. I can honestly tell people that when a bug is reported it is often fixed within 48 hours. That's so amazing that my colleagues generally won't believe me.
3. For students especially it is a great deal. Here is one of the most powerful spatial technologies available and they can keep it legally for their own research.

Unfortunately, because GRASS still is rather clunky on Windows and daunting to install in spite of the great installation web site done by Huidae Cho I have to temper my enthusiasm a bit for Windows users. I'm very hopeful about the efforts by Radim Blazek and others to produce a Windows-native GRASS. Even though I'm a die-hard Mac user, the majority of computer users here live in the Windows world, of course. I've been recommending QGIS to Windows folks a lot especially if they are relatively new to GIS. I'm getting folks interested in Linux and we'll see how that goes. For Mac users, Lorenzo Moretti's binaries have made it very easy to recommend GRASS even for less technical users.

What do you think is great about GRASS and which feature is missing?

The raster tools are great. But NVIZ always blows people away the most. I've been working with the true 3D volumetric tools and NVIZ visualisation and this blows people's socks off.

Good management tools for attribute data is the biggest lack that I can think of. As mentioned before, I'd also like a Windows version and better installation for Windows (they need a simple downloadable exe file installer). I've tried to improve the user interface which is important. If people can't figure out how to use a piece of software, its abilities are wasted.

How long have you been engaged in Open Source/GRASS development?

In late 2003, as I was getting to know the program, I wrote (what I thought was) a tactful email to one of the development team (maybe Markus) about the lamentable state of the tcltkgrass GUI. It was way out of date, not very well organised, missing half of the commands, and had several commands in twice. He tactfully wrote back that if I was concerned about the GUI, I could volunteer to fix it. I didn't write again for awhile. But in early 2004, after returning from sabbatical, I decided I'd take up the challenge and at least try to get all the commands in the GUI. I wanted to do this so I could use GRASS for my spatial technologies course. I added about 200 commands to tcltkgrass for GRASS 5.3 and reorganised the menu to make it (IMHO) easier to navigate. I also wrote my first script. There was no easy way to get a nice GRASS display into another program (ps.map is powerful but easy it is NOT). So with considerable help from others, I wrote d.out.png. GRASS 5.1-5.7 development was underway at the time, and there was no overall interface to speak of for the new version. So I promised Markus that I'd work on it next. Radim had ported the display manager from 5.3 to 5.7 and Markus got me started on a menu. The first thing I had to do was find out which of the 400+ commands had been ported from 5.3 to 5.7. So I made a little text file of the 5.3 commands, so I could note whether they had or had not been ported to 5.7 and check off when I did the tcltkgrass menu for the command. (This BTW was the origin of the GRASS 6 porting list on the WIKI site).

Since early 2004, I've gone from nervously modifying existing TclTk menu files to finally learning enough TclTk to reprogram the entire GUI. This is the fun and challenging part working on an open source project like this. I've also learned enough shell-scripting to go from the first tentative d.out.png to d.vect.thematic that I finished at the beginning of 2005. This gave me the skills to write a very useful script that will compute USPED (Unit Stream Power Erosion/Deposition) for my NSF research project. I started my graduate students working with this and they have now turned this into a full-fledged landscape modeling platform. It's also rewarding to think that I may be helping a broad and diverse group of fellow scientists and GIS users around the world. Finally, I've enjoyed collaborating with GRASS pen-pals' around the globe. It's a great group of people to work with.

You have been developing the Tcl/Tk interface for quite a long time now, what are your plans for the future?

I released version 2 of the GIS Manager earlier

this year to help move GRASS toward a new user interface. It's becoming clear from the UI discussion last year and continuing into this year, that GRASS needs to shed its dependence on x-displays in order for a more sophisticated GUI to be developed. The new GIS Manager is a step in that direction. We are currently planning to replace TclTk with the more powerful wxPython interface toolkit. (Getting a start on learning Python and wxPython was my summer project.) However, in the interim, it is clear that we can get a lot more out of TclTk than we were previously.

Why are you dedicating so much time into GRASS? Philosophy, profession?

It's great and personally rewarding to be able to collaborate with the people on the GRASS team, benefit my own research and teaching, AND have fun at the same time. I think open source is a wonderful philosophy for software development and one that fits very well into an academic environment. We here at ASU are strongly encouraged to make meaningful contributions of our expertise to our community. Being a part of the GRASS project is one way I can do this.

Over the past couple years, I've had a wonderful opportunity to work with some really bright, dedicated, and diverse people to help produce some of the best geospatial software that money can buy. Then I get to turn around and give it to students and researchers for free. I enjoy that. But there is a more serious, underlying set of motivations that I have. As someone whose research interests centre around the long-term outcomes of our interactions with the environment, it is clear to me that these interactions are fundamentally critical for ensuring that our children have a world worth inhabiting. The past has many lessons of successes and failures in which the way in which people interacted with the world around them had important consequences for their well-being and their social and physical survival. But whereas, in the past a society could succeed or fail and have minimal impact on others, now, due to our highly connected, global social and information network, such successes and failures have repercussions for all of humanity.

However, socio-environmental interactions are highly complex and even more so in our as they reach global scales—beyond even our extraordinary human abilities to understand complex systems. Yet understand them we must or we put our future at risk. We have become a keystone species in most terrestrial ecosystems, leaving us no choice but to manage these systems for better or for worse.

So what does this all have to do with GRASS? New kinds of computer technology and data never before available (from ancient climates to satellite im-

agery) give us the tools to understand the operation of coupled human-natural systems (I use the term socioecosystems) better than we ever could before. This kind of understanding gives us a better chance to make more intelligent choices about how we manage our interactions with the world. But because the emergent properties of these highly complex systems produce many surprises in spite of the potential for better understanding, we stand a much better chance if we can come up with a high diversity of potential solutions to problems as they arise.

By making complex geospatial analysis and modelling technology as widely available as possible—from undergraduate students in Arizona, to environmental planners in Italy, to ecologists in Sri Lanka, to agronomists in South Africa—we give ourselves a better chance of finding diverse solutions to old and new problems, and so making this a better world for our children. As a citizen of a 'first world' country, I'm very fortunate to have a life that allows me the opportunity to spend time on something like the GRASS project. In working on the GRASS project, I have an opportunity to give back of my good fortune to others and at the same time help 'hedge our bets' for the future by making tools to better understand our world available to all who need them.

What would be in your opinion a major breakthrough for GRASS?

Seamless 2D-2.5D-3D GIS and visualisation. GRASS is already way ahead of the competition in this direction. Integrating these abilities into a single way of analysing and visualising spatial data would make GRASS THE outstanding program in its class by far.

When do you think will this be accomplished? Will it be part of the new UI design? Any time-frame?

I hope soon. Glynn Clements and others on the development team have been working to bring the internal GRASS display architecture into the 21st century over the past few months. Recent changes to NVIZ have freed it from needing an xterminal and I'm now working with Bob Covill to update the TclTk side of this module. Hopefully, these changes will make it easier to integrate NVIZ-like displays with the rest of GRASS visualisation as the display architecture and GUI evolve. In the new GIS Manager, I've set it so that you can create an NVIZ display (using current NVIZ technology) from the same set of layers that create your 2D display in a map display window. This is not the kind of integration I envision ultimately for GRASS. But perhaps it will give some inspiration to move in that direction. One recent development that gives me cause for optimism is the inclusion of GRASS as a founding member of



the new Open Source Geospatial Foundation (OSGeo). This could provide resources, code, and inspiration for GRASS development to take place much more rapidly than when it was on its own. This is an

exciting development for geospatial technologies in general.

Thanks for the interview and all the best for your GRASS engagement!

The GRASS Development Team announces GRASS GIS 6.2.0

released 31 Oct 2006

We are happy to announce that a new stable version of GRASS GIS has been released today. This release adds hundreds of new features, support for the latest GIS data formats, and includes new translations for many languages. The Geographic Resources Analysis Support System, commonly referred to as GRASS, is a Geographic Information System (GIS) combining powerful raster, vector, and geospatial processing engines into a single integrated software suite. GRASS includes tools for spatial modeling, visualization of raster and vector data, management and analysis of geospatial data, and the processing of satellite and aerial imagery. It also provides the capability to produce sophisticated presentation graphics and hardcopy maps. GRASS is currently used around the world in academic and commercial settings as well as by many governmental agencies and environmental consulting companies. It runs on a variety of popular hardware platforms and is Free open-source software released under the terms of the GNU General Public License. Joining GRASS's well-developed raster engine, the GRASS 6 series introduced a new topological 2D/3D vector engine featuring support for vector network analysis and SQL-based DBMS management of linked attributes. This new release improves the integration and functionality of the raster and vector engines, and greatly enhances 3D raster volume (voxel) support. Additionally, this release debuts a new graphical GIS manager and menu system, while an improved version of the old GUI display manager has been retained

for legacy support. The NVIZ visualization tool has been enhanced to display 3D vector data and voxel volumes, and now supports the creation of on-the-fly MPEG animations. Further improvements include substantial message translations (i18n) with support for FreeType fonts, including multi-byte Asian characters, and the inclusion of tools to create new project locations automatically given a georeferenced data file or EPSG code. This is the first release of GRASS as a proposed founding project of the new Open Source Geospatial Foundation. In support of the movement towards consolidation in the open source geospatial software world, GRASS is tightly integrated with the latest GDAL/OGR libraries. This enables access to an extensive range of raster and vector formats, including OGC-conformal Simple Features. GRASS also makes use of the highly regarded PROJ.4 software library with support for most known map projections and the easy definition of new and rare map projections via custom parameterization.

Platforms supported by GRASS

GNU/Linux, Mac OS X/Darwin, Microsoft Windows (native using MinGW or with full UNIX support via Cygwin), Sun Solaris (SPARC/Intel), Silicon Graphics Irix, HP-UX, DEC-Alpha, AIX, BSD, iPAQ/Linux and other UNIX compliant platforms. GRASS runs on both 32 and 64 bit systems with large files (>2GB) supported by many key modules.

Software download/CDROM

- <http://grass.itc.it>
- <http://grass.ibiblio.org>
- numerous mirror sites
- GRASS on CDROM/DVD

The new source code is available now and binary packages for major operating systems will be published shortly.

For details on GRASS software capabilities please refer to: <http://grass.itc.it/intro/general.php>, the previous GRASS 6.0.0 Announcement, and the newly renovated Wiki collaborative help system.

What's new in GRASS 6.2.0

(selected improvements)

- Numerous bug fixes
 - see the ChangeLogs (6.1, 6.2) for details
- Source code quality/libraries:
 - The GRASS code base is now in large part ANSI C compliant
 - Ported natively to MS-Windows (MinGW based)
 - Source code header files: improved, many compiler warnings fixed
 - Compilation: compatible with GCC 4.x
 - Programmer's Manual: continued Doxygen integration and automated generation into PDF and HTML formats. Publicly available for download and perusal.
 - Improved policies for code submission specified in the SUBMITTING files
 - GRASS-SWIG prototype interface added (library bindings for Perl and Python)
 - DBMI: SQLite driver added; SQL parser extended (support for expressions, new types, etc.)
 - DBMI: MySQL driver rewritten; MeSql added
 - Support for long map/mapset names
 - Raster maps: ZLIB compression bug fixed for tiny maps
 - Raster maps: optional large file support (LFS, experimental) for maps > 2GB
 - Display: X11/PNG driver rewritten, added RGB-raster operations

- Display: Movement toward X-Windows as an optional dependency (new multi-platform GUI support)
- Display: High CPU use during interactive mouse functions fixed
- Projection code database: updated to EPSG 6.11.2
- Tcl/Tk 8.4 support for Debian and other platforms using threaded libraries
- Full FFTW3 support for fast Fourier transforms
- NetBSD configuration fixes
- New quality control systems:
 - New internal GRASS test suite (scripts collection in "testsuite/")
 - New external GRASS test suite (TU Berlin)
 - New external GRASS Quality Assessment and monitoring system (École Polytechnique de Montréal and ITC-irst)
 - CVS-commit reports into IRC '#grass' channel via CIA - The open source informant
- Graphical User Interface (GUI):
 - All modules: Major improvements in the auto-generated GUIs
 - **gis.m**: NEW - GIS manager added as a replacement for **d.m** (optional)
 - **d.m**: Legacy support provides access to the latest features while preserving code maturity
 - QGIS integration: fixes for the GRASS plugin and toolbox available from Quantum GIS
 - Continued behind-the-scenes infrastructure refinements for the next generation GUI
- GRASS Extensions Manager (GEM): NEW
 - Configure, compile and install additional GRASS modules without needing the GRASS source code. Simplifies the addition of new modules or themed module groups.
- Modules/Scripts:
 - Message translation (i18N): added and extended to more than a dozen languages, Tcl/Tk GUI and shell script messages are now translatable

- Documentation/man pages: various fixes and improvements (more examples added, including graphics, improved style, new introductory pages)
- Raster modules: improved support for meta-data and map history
- **d.correlate**: NEW - create a graph of the correlation between data layers
- **d.graph**: rewritten and extended, merged with d.mapgraph; support for symbols
- **d.grid**: added support for geographic grid overlay on non lat/lon map projections and display of coordinate values
- **d.labels, v.label**: d.paint.labels renamed d.labels; significant fixes and improvements to the label subsystem
- **d.m**: improved layout; added functionality
- **d.menu**: NEW - creates and displays a menu within the active graphics monitor (tool for interactive scripts) (port from GRASS 5)
- **d.mvmon**: NEW - moves displayed maps to another monitor
- **d.out.file**: NEW - saves active display monitor graphics to an image file (PNG, JPEG,...)
- **d.out.gpsdrive**: NEW - exports display monitor to a GpsDrive compatible back-drop image
- **d.polar**: NEW - draws a polar diagram for an angle map such as topographic aspect or flow direction
- **d.rast.arrow**: many enhancements, support for magnitude as well as 360 degree directional inputs
- **d.text**: added support for text rotation
- **d.vect**: variable vector line width added, random colors for points and lines, dynamic width and colors from attribute data
- **d.vect.thematic**: NEW - customizable thematic mapper for vector map displays
- **g.html2man**: fixes for improved man page output
- **g.transform**: NEW - utility to compute coordinate transformations based upon GCPs, including error analysis
- **gis.m**: NEW alternative graphical GIS manager (see above)
- **i.in.spotvgt**: NEW - import SPOT-VGT NDVI satellite imagery into a raster map
- **i.landsat.rgb**: NEW - auto-enhancement of colors for LANDSAT imagery
- **i.ortho.photo, i.rectify**: no longer signal completion with email notification
- **i.spectral**: fix for finding gnuplot
- **i.points, i.vpoints**: various fixes, including reverse transform overlay of vector maps from the target projection
- **m.proj**: NEW - utility to convert coordinates from one projection to another (automated frontend for cs2cs)
- **NVIZ**: integrated into single user interface; animation labels; new fly-through navigation; direct output of animations to MPEG with FFMPEG library; menus polished; full Tcl/Tk 8.4 support
- **ps.map**: many improvements (extended RGB support, etc), new vector fill patterns including vector legend support
- **r.carve**: NEW - hydrologic module for transforming vector stream data into a raster map, including the subtraction of stream depth from the output DEM
- **r.flow**: block erroneous Lat/Lon calculations
- **r.in.srtm**: support for US 1-arcsec tiles
- **r.in.wms**: NEW - download and import data from WMS servers
- **r.in.xyz**: NEW - creates a raster map from an assemblage of many coordinates using univariate statistics (LIDAR/Swath bathymetry import tool)
- **r.lake**: NEW - raises lakes in a DEM from a seed at a given water level
- **r.le**: many stability fixes
- **r.mapcalc, r3.mapcalc**: acos(), asin(), pow(), &&& and ||| added for more intuitive handling of null data
- **r.mask**: NEW - create a MASK for limiting raster operations (port from GRASS 5)
- **r.out.gdal**: added support for multiple CREATEKEY and METAKEY parameters
- **r.out.vtk**: NEW - converts raster maps into the VTK-ASCII format for 3D visualization
- **r.profile**: improved handling of input data
- **r.sim.sediment**: NEW - overland flow hydrologic erosion model based on duality particle-field concept (SIMWE)

- **r.sim.water:** NEW - overland shallow water flow hydrologic model based on duality particle-field concept (SIMWE)
- **r.sun:** fixed units in description; new shadow algorithm; correction factor for shadowing to account for the Earth's curvature
- **r.support:** NEW - raster map layer support module (port from GRASS 5)
- **r.thin:** fixed maximum iterations for lines
- **r.to.rast3:** NEW - converts 2D raster map slices to one 3D raster volume map
- **r.to.vect:** add flag to not build topology (for *r.in.xyz + v.surf.rst*)
- **r3.*:** many code improvements and extended functionality for 3D raster volume (voxel) data
- **r3.cross.rast:** NEW - Creates a cross section 2D raster map from a 3D raster volume map based on a 2D elevation map
- **r3.out.vtk:** NEW - converts 3D raster maps into the VTK-ASCII format for visualization
- **v.centroids:** NEW - adds missing centroids to a vector map's closed area boundaries
- **v.db.addcol:** NEW - creates one or more columns in the attribute table connected to a vector map
- **v.db.addtable:** NEW - creates a new attribute table and connects it to an existing vector map
- **v.db.droptable:** NEW - removes an existing attribute table from a vector map
- **v.db.reconnect.all:** NEW - reconnect vector maps to a new database
- **v.db.select:** multiple column support added
- **v.db.update:** NEW - allows the user to assign a new value to a column in an attribute table
- **v.drape:** NEW - converts a 2D vector map into a 3D vector map by sampling an elevation raster
- **v.digit:** new layout, tools added, bugs fixed
- **v.dissolve:** NEW - dissolve boundaries between adjacent areas in a vector map
- **v.external:** GID search added; auto-search of FID added
- **v.extrude:** NEW - extrudes flat vector objects into 3D vector faces of a defined height
- **v.in.dxf:** NEW - rewritten for GRASS 6, support for 3D features added
- **v.in.ogr:** repaired the `--overwrite` flag, selective import with WHERE option
- **v.in.e00:** various fixes
- **v.in.gns:** NEW - imports US-NGA GEONet Names Server (GNS) country files
- **v.in.gpsbabel:** NEW - import Waypoints, Routes, and Tracks from a GPS receiver or a GPS ASCII file into a vector map (supports many GPS formats)
- **v.in.mapgen:** NEW - imports Mapgen or Matlab vector maps (port from GRASS 5)
- **v.kernel:** speed improvements
- **v.lrs.*:** NEW - Linear Reference System for vector line networks
- **v.out.ogr:** 3D 'face' export added (3D vectors, for Google Earth KML format, etc.)
- **v.out.vtk:** NEW VTK vector export
- **v.patch:** attribute transfer, if table structures are identical
- **v.rast.stats:** NEW - calculates univariate statistics from a GRASS raster map based on vector objects
- **v.report:** NEW - calculates and reports geometry statistics for vector maps
- **v.surf.rst:** default `npmin` option fixed; fix for reading spatially variable smoothing parameter
- **v.to.rast:** added option to compute the direction (angle) of lines
- **v.to.rast3:** NEW - converts a vector points map into a 3D raster map
- **v.univar.sh:** NEW - calculates extended univariate statistics on a selected vector map attribute table column
- **v.what.vect:** NEW - uploads the values of a given vector map into a coinciding vector point map's attribute table
- Misc. display modules: Increased support for true-color RGB decorations
- Misc. vector modules: Support for dynamic fill color and dynamic feature sizing from DB attribute columns
- Scripts: Fixes for awk and related calculation problems for various locales
- Scripts: Variable test portability issues fixed

For a comprehensive list of changes to modules see the 6.1 and 6.2 ChangeLog files.

For a complete list of commands available in GRASS 6.2.0 see the online manuals.

We are always looking for testers, code developers, and technical writers to help us maintain and accelerate the development cycle.

The GRASS GIS project is developed under the terms of the GNU General Public License (the GPL) in the open by volunteers the world over.

GRASS differs from many other GIS software packages used in the professional world in that it is developed and distributed by users for users, mostly on a volunteer basis, in the open, and is given away for free.

Emphasis is placed on interoperability and unlimited access to data as well as on software flexibility and evolution rate.

Release history:

- GRASS GIS 6.1.0 technology preview released 11 August 2006
- GRASS GIS 6.2.0beta1 released 28 August 2006
- GRASS GIS 6.2.0beta2 released 30 August 2006
- GRASS GIS 6.2.0beta3 released 18 September 2006
- GRASS GIS 6.2.0RC1 released 26 September 2006
- GRASS GIS 6.2.0RC2 released 6 October 2006
- GRASS GIS 6.2.0RC3 released 24 October 2006
- GRASS GIS 6.2.0 released 31 October 2006

Please read the following article about the GRASS 6.2.1 release for further enhancements and bug fixes.

The GRASS Development Team announces GRASS GIS 6.2.1

released 12 Dec 2006

We are happy to announce that a new stable version of GRASS GIS has been released today. This release fixes several bugs discovered in the 6.2.0 source code. It is solely for stability purposes and adds no new features. Besides bug fixes it also includes a number of new message translations, updates for the help pages, and will better handle errors caused by missing or incorrectly installed support software. It also introduces a new 3D raster module which was left out of the last release due to time constraints.

What's new in GRASS 6.2.0

(selected improvements from the more than 100 minor and important fixes)

- System and Libraries:
 - Install of include files problem on Solaris (Glynn)
 - Handle non-standard ETRS_1989 datum name (Paul)
 - Graphical User Interface (GUI):
 - Fixes for geo-rectifier (Michael)
 - Zooming fixes (Michael)
 - Meaningful error messages on failed startup (Michael)

• Modules/Scripts:

- d.histogram: clear just the current frame, not the full screen (Hamish)
- i.group: fix subgroup listing (Hamish)
- ps.map: broken for named paper sizes (Hamish, Glynn)
- v.db.select: fix errors with the SQL where= option (Hamish)
- v.in.db: don't copy the entire attribute table with a SQL where= query (Martin)
- v.in.ogr: fix for gcc4.1.x and non-C locale (Andrey Kiselev)
- r.to.rast3elev: **NEW** - Completes the 3D raster tool set. Creates a 3D volume map based on 2D elevation and value raster maps.

For a comprehensive list of changes to modules see the 6.1, 6.2.0 and 6.2.1 ChangeLog files.

For a complete list of commands available in GRASS 6.2.1 see the online manuals.

Release history:

- For earlier releases see previous article.
- GRASS GIS 6.2.1RC1 released 6 December 2006
- GRASS GIS 6.2.1 released 12 December 2006

Editor-in-Chief:

Martin Wegmann

DLR – German Aerospace Center @

Remote Sensing and Biodiversity Unit

Dept. of Geography, University of Würzburg, Germany

BIOTA-Project

[http://www.geographie.uni-wuerzburg.de/
arbeitsbereiche/fernerkundung/](http://www.geographie.uni-wuerzburg.de/arbeitsbereiche/fernerkundung/)<http://www.biota-africa.org>wegmann_at_biozentrum.uni-wuerzburg.de**Editorial Board:**

Paul Kelly, Markus Neteler and Martin Wegmann

Editor News & GRASS related articles:

Andrew Davidson

Language Editor:

Ondrej (Andy) Mitas

The *GRASS/OSGeo newsletter* is a publication of the *GRASS* and *OSGeo Foundation*. The base of this newsletter, the $\text{\LaTeX}2_{\epsilon}$ source has been kindly provided by the R News editorial board (<http://www.r-project.org>). All articles are copyrighted by the respective authors. Please use the GRASS newsletter url for submitting articles, more detailed concerning submission instructions can be found on the GRASS homepage (<http://grass.itc.it> and its mirrors).

OSGeo Homepage: <http://www.osgeo.org>GRASS Project Homepage: <http://grass.itc.it/>

Newsletter online:

<http://grass.itc.it/newsletter/index.php>**Acknowledgements**

Various reviewers & the R News Project

ISSN 1614 - 8746