OSGeo Journal Volume 8

February 2011

FOSS4G 2009
Conference Proceedings

OSGeo Community
News & Announcements
Case Studies
Integration Examples

The Annual International

FREE & OPEN SOURCE SOFTWARE FOR GEOSPATIAL

Conference Event

2011.FOSS4G.ORG

OSGeo
Your Open Source Compass

Volume 8 Contents

From the Editor

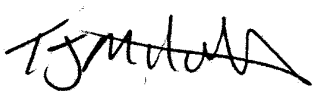
OSGeo has just past its 5th birthday, along with this 8th volume of the OSGeo Journal! With this edition we bring a few news headlines from the past couple months, a few general articles and, most significantly, several top papers from the **FOSS4G 2009** conference event held in Sydney, Australia.

The Journal has become a diverse platform for several groups and growth in each area is expected to continue. The key groups that read and contribute to the Journal include software developers sharing information about their projects or communities, power users showing off their solutions, academia seeking to publish their research and observations in a peer-reviewed, open source friendly medium. OSGeo also uses the Journal to share community updates and the annual reports of the organisation.

Welcome to those of you who are new to the OSGeo Journal. Our Journal team and volunteer reviewers and editors hope you enjoy this volume. We also invite you to submit your own articles to any of our various sections. To submit an article, register as an "author" and sign in at `http://osgeo.org/ojs`. Then when you log in you will see an option to submit an article.[1]

We look forward to working with, and for, you in the upcoming year. It's sure to be an interesting year as we see OSGeo, Open Source in general and all our relate communities continue to grow. Nowhere else is this growth more apparent than at our annual conference: **FOSS4G 2011 Denver**, September, 2011.[2] Keep an eye on your OSGeo mailing lists, blogs and other feeds to follow the latest FOSS4G announcements, including the invitation to submit presentation proposals.[3] It will be as competitive as ever to get a speaking slot, so be sure to make your title and abstract really stand out.

Wishing you the best for 2011 and hoping to see you in Denver!



Tyler Mitchell
`tmitchell@osgeo.org`
Editor in chief, OSGeo Journal
Executive Director, OSGeo

[1] The direct URL for article submission is: `https://www.osgeo.org/ojs/index.php/journal/author/submit`
[2] FOSS4G 2011 Denver: `http://2011.foss4g.org`
[3] FOSS4G 2011 Abstract Submission: `http://2011.foss4g.org/program`

Integration Examples

Exporting Geospatial Data to Web Tiled Map Services using GRASS GIS

Tomáš Cebecauer and Marcel Šúri

Abstract

We present a method for exporting raster-based geospatial data to the web environment of the Tiled Map Services with a focus on Google Maps and Microsoft Virtual Earth. The method has been implemented in the open source GRASS GIS software, and it includes the exact re-projection of raster data, their tiling, and export to the hierarchical structure of PNG graphical files. The approach is based on the adaptation of projection parameters of the standard PROJ4 library, and implementation of the technical specifications and tiling scheme of the services. The whole procedure was wrapped into the GRASS GIS command *r.out.gmap*.

Keywords: Geographic Information Systems, Internet, Raster data, Web Tiled Map Services, Google Maps

Introduction

Until a few years ago, the only way to share dynamic maps over the Internet was using dedicated Web Map Services (WMS), such as the open source MapServer or ArcIMS by ESRI. These WMS provide high levels of interactivity with users and the possibility to generate maps on-demand from the underlying geographical information. While WMS provide relatively complex functionality, the concept of on-the-fly map generation demands a lot of data processing and image manipulation, thus resulting in a slow response to user requests. Due to high load on the server resources, the WMS applications have remained the domain of a limited number of internet map service providers.

In the last years, web portals such as Google Maps (GM), Microsoft Virtual Earth (MVE), Yahoo! Maps, and OpenLayers have pioneered an era of interactive map handling and sharing over the Internet, providing tools for geographical search and browsing global datasets such as topographic maps, satellite imagery and terrain. These portals have gained high popularity thanks to simple use, fast response and an easy way to build customised applications. The high performance is achieved by a change of the underlying concept from dynamic generation of complete maps to the use of static pre-rendered raster images, and reduction of dynamically generated content to simple vector or raster overlays. As the pre-rendered images are usually stored in a hierarchical system of image tiles these services are called Tiled Map Services (TMS)[19].

Customized client applications with tailored functionality for TMS can be built using Application Programming Interfaces (API) that provide tools also for integration of user-prepared maps. Although having reduced functionality compared to WMS, the TMS APIs have opened an opportunity to share maps over the Internet without a need of specific server based WMS technology. A simple TMS may be established only by storing the pre-rendered tiles in PNG or JPEG format on the web server without employing additional applications. Using API from GM or MVE provides benefits of the direct access to the global coverage of geographical data, maps, and to the powerful search engines.

Although it is relatively simple to set up a cus-

[19] Tile Map Service Specification., v1.0, OSGeo, 2007. http://wiki.osgeo.org/index.php/Tile_Map_Service_Specification

tomised TMS, the generation of image tiles is not straightforward. Spatial reference of the user's thematic maps most often differs from the projection and tiling schema required by TMS, and the data have to be re-projected and segmented accordingly. Most solutions for custom data tiling available on the Internet are based on the use of simple web tools or graphical editors, where map projection is overlooked or simplified to raster rubber-sheeting. These approaches may result in significant positional distortions, especially when dealing with global or continental databases (Figure 1).

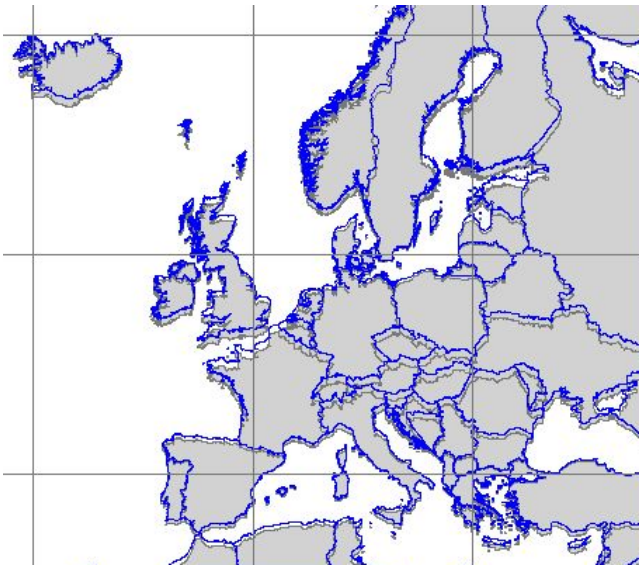


Figure 1

We present an approach, developed for the open source GRASS GIS software, for exact (pixel-by-pixel) map re-projection, segmentation and conversion of raster data to image tiles for TMS with focus on GM and MVE environment.

Data Tiling in Google Maps and Microsoft Virtual Earth

To allow fast display of maps at any zoom detail, a hierarchical and positional relation between the TMS image tiles is established. Both GM and MVE use the recursive division of square tiles (Figure 2). The tiles are divided by a factor of two in each direction, thus while zooming to a more detailed view the original tile is at the successive zoom level replaced by four tiles at higher resolution. The global view in GM is represented by the zoom level 0 that is composed by one tile of 256 x 256 pixels size. The zoom in MVE starts with four tiles at the zoom level 1.

The area represented by 1 pixel is determined by the number of divisions of the Earth globe in the given zoom level. For a pixel at the equator, the resolution for each zoom level, *reszoom* can be calculated as follows:

$$res_{zoom} = \frac{2\pi R_{sph}}{tile\_size.2^{zoom}} \quad (1)$$

where $tile\_size$ is the size of a tile in pixels in x direction, $zoom$ is the zoom level and R_{sph} is the Earth radius at the equator. The recursive division increases the resolution quite rapidly: raster spatial resolution of approx. 100 m is reached between the zoom levels 10 and 11, and resolution of 1 m is close to the zoom level 17.

Although tiling in GM and MVE is the same, the tile numbering differs. GM tiles use three numbers for the identification of a tile: zoom level, row and column of the tile. The tiles are ordered left to right and top to down with index starting from 0 for upper left tile. The MVE tile system uses the quadtree approach[20], where each quadrant is labelled by a number from 0 to 3. At the subsequent zoom level the quadrant number is appended at the end of the "ancestor" quadrant number (see Figure 2). A similar system is used by GM for the IDs of satellite image tiles, but numbers are replaced by letters "*q, r, s, t*"[21].

Map Projection

The spatial reference of GM and MVE is set to the Mercator map projection[22]. Mercator belongs to the group of cylindrical map projections, which means that in a normal position all meridians are parallel with y-axis of the coordinate system, and at the same time meridians are perpendicular to the parallels. The projection is conformal, thus preserving angles and shapes of small objects. The main drawback is the area distortion that increases with distance from equator towards the poles, where it goes to infinity. This results in exaggeration of objects close to poles, so Greenland seems to be slightly larger than South America (actually it is eight times smaller).

The implementation of the Mercator map projection in GM and MVE introduces several modifications. The most important is a use of the spherical form of the projection, which is defined by the general equations (Snyder, 1987):

$$\begin{aligned} x &= R_{sph}(\lambda_{sph} - \lambda_0) \\ y &= R_{sph}\frac{1}{2}ln\left(\frac{1 + sin\varphi_{sph}}{1 - sin\varphi_{sph}}\right) \end{aligned} \quad (2)$$

[20]MSDN. Virtual Earth Tile System http://msdn2.microsoft.com/en-us/library/bb259689.aspx
[21]MAPKI. Satellite Tile Layout. http://mapki.com/wiki/Satellite_Tile_Layout
[22]Google Maps API Reference. http://code.google.com/apis/maps/documentation/reference.html

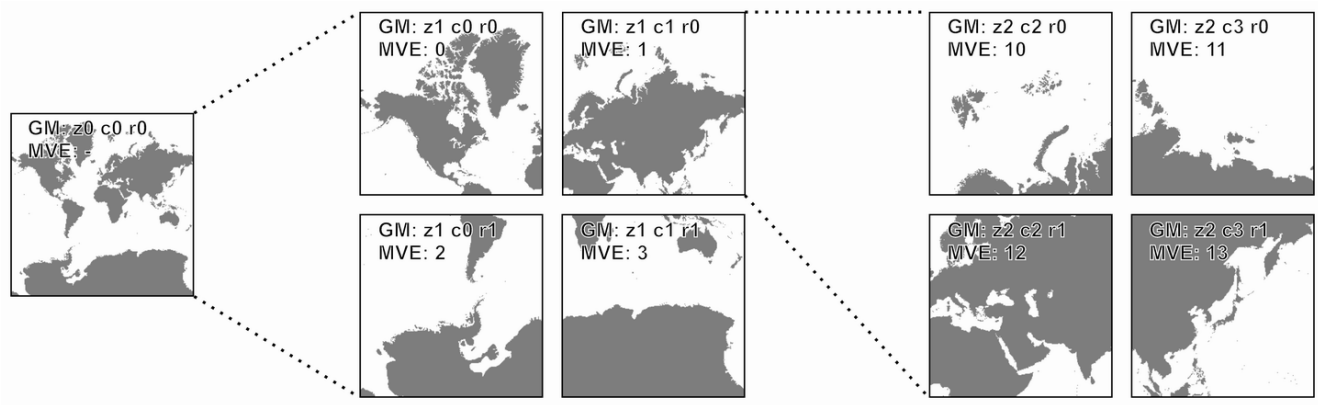


Figure 2

where λ_{sph} and φ_{sph} represent respectively longitude and latitude, λ_0 is the central meridian and R_{sph} is a radius of the Earth in spherical datum. The use of spherical form simplifies the underlying calculations and accordingly it results in about 0.33% scale distortion in the y-axis direction which is not noticeable for visual applications.

Using the notation of the PROJ4 library, the spherical Mercator projection is defined by the following set of parameters:

+proj=merc +lat_ts=0 +lon_0=0 +k=1.0 +x_0=0 +y_0=0 +a=6378137.0 +b=6378137.0 +units=m

Data re-projection using PROJ4 with this set of parameters does not align results to the GM or MVE maps (see Fig. 3), because it uses standard datum transformation of coordinates between the WGS84 ellipsoid and the sphere with the radius 6378137 m.

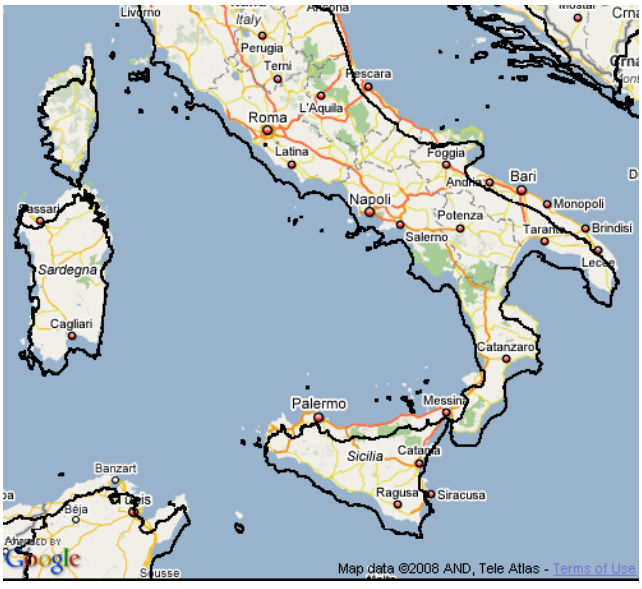


Figure 3

The projection implementation in GM and MVE bypasses this transformation and assumes that the coordinates on the sphere equal those on the ellipsoid. This even more simplifies the calculations as there is no need of any datum transformation and the WGS84 ellipsoidal latitude/longitude coordinates are used for the spherical Mercator projection. Then, the general form of the projection in eq. 2 can be rewritten as follows:

$$x = R_{sph}(\lambda_{wgs84} - \lambda_0)$$
$$y = R_{sph}\frac{1}{2}ln\left(\frac{1 + sin\varphi_{wgs84}}{1 - sin\varphi_{wgs84}}\right) \quad (3)$$

where λ_{wgs84} and φ_{wgs84} are longitude and latitude in the WGS84 datum, whereas R_{sph} is the radius of the Earth in the spherical datum (set to 6378137.0 m). Assuming λ_0 equal to 0, the equation for x might be even more simplified. However, such projection is not easy to implement in the coordinate transformation packages, as they were created with the intention of precise cartographic transformations. Luckily the PROJ4 library provides a solution by tricking the transformation using the @null grid shift file:

+proj=merc +lat_ts=0 +lon_0=0 +k=1.0 +x_0=0 +y_0=0 +a=6378137.0 +b=6378137.0 +units=m +nadgrids=@null

The implementation of the Mercator projection for displaying data in the square tiles limits the parts of the Earth that can be shown to between approximately +/-85.05 degrees of latitude. This overcomes the problem of the y coordinates approaching the infinity at the poles.

Another specific feature of the GM map projection is the use of pixels as output units. As the pixels are not fixed and their number increases with zooming in, the projected x and y coordinates should be rescaled to fit the pixel units of individual zoom level. This can be done by dividing the coordinates with pixel resolution for a specified zoom level derived in eq. (1).

$$x_{px,zoom} = xres_{zoom}$$
$$y_{px,zoom} = yres_{zoom} \quad (4)$$

and after simplification:

$$x_{px,zoom} = \frac{tile\_size.2^{zoom}}{2\pi}(\lambda_{wgs84} - \lambda_0) \quad (5)$$
$$y_{px,zoom} = \frac{tile\_size.2^{zoom}}{2\pi}\frac{1}{2}ln\left(\frac{1+sin\varphi_{wgs84}}{1-sin\varphi_{wgs84}}\right)$$

Eq. (5) actually defines the projection for each level individually, which may be useful for implementation in a customized processing. However, in GIS a change of the projection for each zoom level is not practical. A more straightforward solution is the use of a projection definition based on the Earth spherical datum in eq. (3), and to do the rescaling to the pixel units using eq. (4) afterwards. It is worth noting that the pixel coordinates derived using eq. (4) and (5) have the center of the coordinate system placed at the equator and longitude λ_0, usually at 0°. To make the calculations and tiling in the pixel space simpler, it is suggested to move the centre of the coordinate system to the upper left pixel and flip the *y* axis towards south.

Besides the Mercator projection, GM allows the use of other reference systems. By specifying transformation functions between pixels and geographical coordinates in the GProjection interface of the GM API, a user can define the projection that best fits his/her requirements. However, this approach does not enable integration of maps from different projections into one application.

Custom data tiling in the GRASS GIS

The GRASS GIS is an open source system with a powerful set of analytical and modelling tools (Neteler, Mitasova, 2008), and it provides all the basic functionality required for creation of GM or MVE tiles. However, the custom data tiling is not straightforward and to provide a one-command solution, the existing tools have to be integrated following the logic of the GM tiling.

The GRASS GIS follows the approach of using one reference system including a map projection and a coordinate system for the whole project (called LOCATION). All operations are restricted to the use of this reference system. Data that are in a different coordinate system have to be stored separately in a different LOCATION and can be accessed by re-projection. This may be achieved using the *r.proj* or *v.proj* commands – a GRASS GIS implementation of the PROJ4 library. As user data are very unlikely to be stored in the GM projection, the thematic raster map tiling should encompass the following steps:

1. Creation of the GM LOCATION with a projection defined by the following parameters:
   `+proj=merc +lat_ts=0 +lon_0=0 +k=1.0 +x_0=0 +y_0=0 +a=6378137.0 +b=6378137.0 +units=m +nadgrids=@null;`
2. Calculation of tiles required to cover the thematic map at specified zoom level;
3. For each tile re-projection of the raster map from source LOCATION to the GM LOCATION;
4. Tile export using predefined naming convention.

We have integrated steps 2 to 4 into a new GRASS command *r.out.gmap* that hides the tiling, re-projection and coordinate rescaling processes from the user, the example:

```
r.out.gmap input=dem location=wgs84
mapset=mydata zoom=5 outdir=gmdem
```

The user only sets a name of the input map, mapset and location, zoom level and name of the directory to export the tiles to. On the output he gets the PNG tiles which can be directly placed on the web server. In such a way it is relatively easy to automate the map publishing for the GM or MVE from whatever GRASS GIS project.

Application – PVGIS web portal

PVGIS (Photovoltaic Geographical Information System, see Šúri, et al. (2005)) is an interactive map-based web system offering free access to geographic data and tools used for performance assessment of solar photovoltaics (Fig. 4) for Europe, Africa, and South-West Asia. In PVGIS we integrated Google Maps API (coded in JavaScript) with our geospatial database and interactive server applications (controlled by PHP). When upgrading the older interface to use Google Maps, of key benefits for the users were the search tool, and the intuitive navigation from the continental to regional levels.

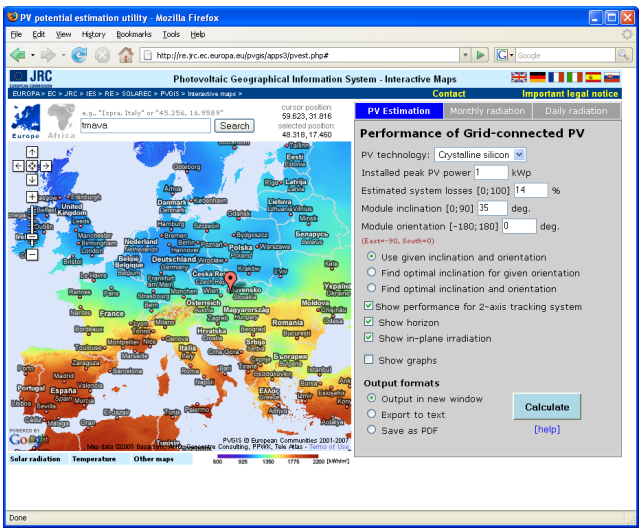


Figure 4

Using the *r.out.gmap*, we have converted a set of GRASS raster data into Google Maps, thematically encompassing solar radiation, temperature, land cover and shaded terrain. The pixel-by-pixel re-projecting makes it possible to keep positional consistency between the Google standard maps, and the PVGIS custom maps. To give an example of the accuracy, the user-selected position in Europe should match with SRTM-3 digital elevation model (van Zyl, 2001), and derived local terrain horizon, both used in a simulation of solar radiation.

Conclusion

We have presented a GRASS GIS approach integrated to a single command for exact re-projection of raster data, and their tiling, and export, according to the requirements of Google Maps and Microsoft Virtual Earth. This approach includes necessary modification of the projection parameters in the PROJ4 library to match the requirements of Tiled Map Services, and implementation of the data segmentation and tiling scheme. The output PNG or JPEG tiles can be directly integrated within the user customised map application on the Internet. The Application Programming Interfaces, available for both systems and the presented tool provide an opportunity to effectively communicate any geospatial information via Internet.

Acknowledgment

The authors would like to thank Thomas Huld for linking the Google Maps Application Programming Interface with the PVGIS data and tools, and to Helena Mitasova, and Jaro Hofierka for valuable comments. This work was partially supported by the VEGA Grant Agency (project 1/3049/27), and by the JRC's FP7 framework program (project 13106, SOLAREC).

References

1. Snyder, J.P., 1987. Map Projections - A Working Manual. United States Government Printing Office, Washington DC, USGS Professional Paper 1395, 383 pp.
2. Neteler, M., Mitasova, H., 2008. Open Source GIS: A GRASS GIS Approach. Third Edition. Springer Science + Business Media, New York, 406 pp.
3. Šúri, M., Huld, T., Dunlop, E.D., 2005. PVGIS: A web-based solar radiation database for the calculation of PV potential in Europe. International Journal of Sustainable Energy, 24, 55-67.
4. van Zyl, J.J., 2001. The shuttle radar topography mission (SRTM): A breakthrough in remote sensing of topography. Acta Astronautica, 48, 559-565.

Tomáš Cebecauer
European Commission, Joint Research Centre
Institute for Environment and Sustainability
Renewable Energies Unit, TP 450, via E. Fermi 1
I-21020 Ispra (VA), Italy
Institute of Geography, Slovak Academy of Sciences
Štefánikova 49, 814 73 Bratislava, Slovakia
tomas.cebecauer AT jrc.it

Marcel Šúri
European Commission, Joint Research Centre
Institute for Environment and Sustainability
Renewable Energies Unit, TP 450, via E. Fermi 1 I-21020 Ispra (VA), Italy
marcel.suri@jrc.it

Imprint

Acknowledgements
Daniel Holt, LaTeX magic & layout support
Various reviewers & writers

The *OSGeo Journal* is a publication of the *OSGeo Foundation*. The base of this journal, the LaTeX 2$_\varepsilon$ style source has been kindly provided by the GRASS and R News editorial boards.

Journal online: http://www.osgeo.org/journal

OSGeo Homepage: http://www.osgeo.org

Mail contact through OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8

ISSN 1994-1897

osgeo.org