

Results of Google Summer of Code 2016 at

OSGeo



Google
Summer of Code



OSGeo

Margherita Di Leo

Anne Ghisla

OSGeo GSoC 2016 admins

Join the ML: soc@lists.osgeo.org

what is GSoC?



Google
Summer of Code

From FAQs: “Google Summer of Code is a program that offers **student developers stipends to write code for various open source projects**. We work with many open source, free software, and technology-related groups to identify and fund projects over a **three-month period**”.

Read more at <https://developers.google.com/open-source/gsoc/faq?hl=en>

OSGeo : The Open Source Geospatial Foundation Visit : osgeo.org

OSGeo Projects

Content Management Systems

GeoNode

Desktop Applications

GRASS GIS
gvSIG
Marble
QGIS

Geospatial Libraries

FDO
GDAL/OGR
GEOS
GeoTools
OSSIM
PostGIS

Metadata Catalogs

GeoNetwork
pycsw

Web Mapping

deegree
geomajas
GeoMOOSE
GeoServer
Mapbender
MapFish
MapGuide Open Source
MapServer
OpenLayers

Incubating Projects

Desktop Applications

Opticks

Geospatial Libraries

MetaCRS
Orfeo ToolBox (OTB)

Web Mapping

istSOS
PyWPS
Team Engine
ZOO-Project

Outreach Projects

GeoForAll - Education and Curriculum
OSGeo Live
Public Geospatial Data

OSGeo Community Projects

Geoinformatica
GeoWebCache
MapProxy
pgRouting
Postal Address Geo-Coder
Virtual Terrain Project

What is

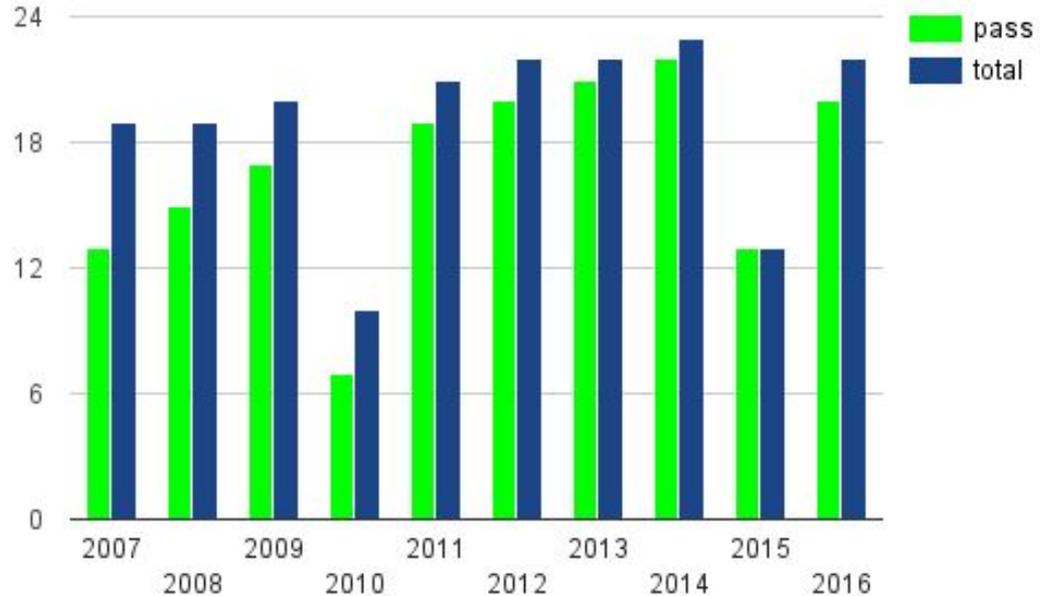


?

GSoC @ OSGeo Stats

Year: pass/total

2007: 13/19
2008: 15/19
2009: 17/20
2010: 7/10
2011: 19/21
2012: 20/22
2013: 21/22
2014: 22/23
2015: 13/13
2016: 20/22



More at http://wiki.osgeo.org/wiki/Google_Summer_of_Code_Application_2017

GSoC 2016 @ OSGeo

22 students accepted

Project: GDAL (1/2)

Student: Avyav Kumar Singh

Mentors: Rob Emanuele, Even Rouault



Title: Introduce Triangulated Surface, Polyhedral Surface and Triangle API in the OGRGeometry core and implement their support in OGR drivers for GDAL

Description: Currently the OGR geometry core of GDAL does not provide support for Triangulated Surfaces, Polyhedral Surfaces and Triangle API. My idea is to introduce these geometries to the OGR core; develop a bare-bones, yet complete API which is flexible enough to be built on top of the existing OGR core and minimal enough to easily enhance; introduce support for select drivers by rewriting the relevant/deprecated parts; test everything I have written rigorously and provide concise documentation for the same.

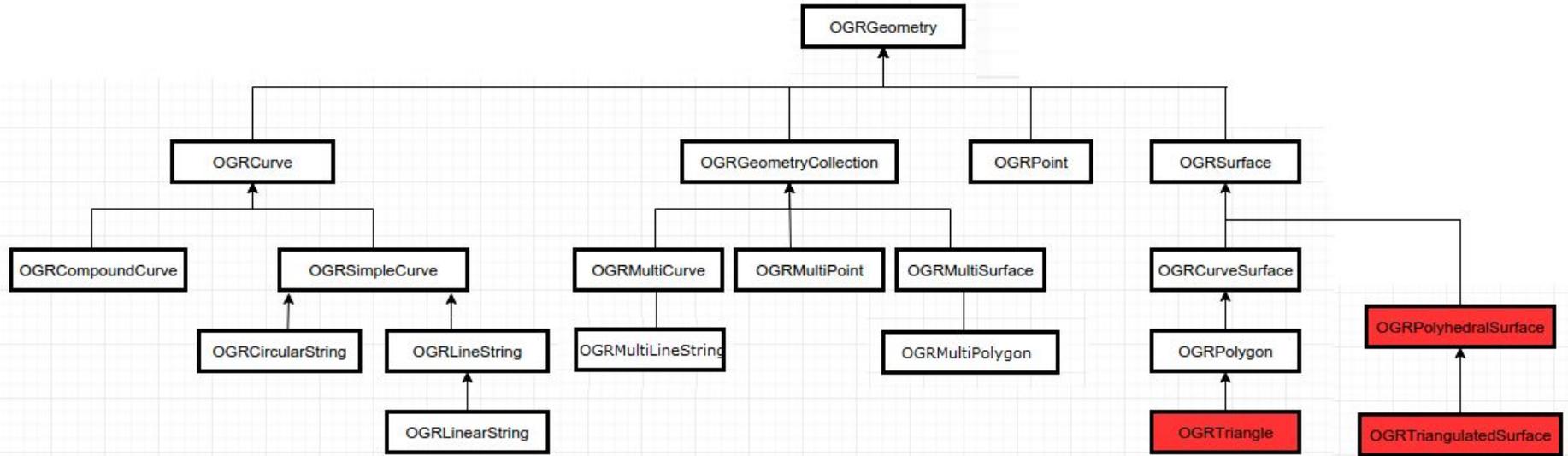
State of the project before GSoC: According to the standard mentioned by OGC 06-103r4, OGR still lacks proper implementation and drivers for a PolyhedralSurface, Triangle and TriangulatedSurface.

Addition to the project:

- Triangle - A subset of polygons, the fundamental difference is that of 3 nodes only and ONLY ONE exterior boundary and NO interior polygons.
- PolyhedralSurface - A 3D figure made exclusively of Polygons.
- TriangulatedSurface - A subset of PolyhedralSurface; a 3D figure which consists exclusively of Triangles.
- Modification to following drivers for OGR - PostGIS, ShapeFile, DXF and GML to work properly (reading + writing capabilities) for the new geometries.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003506.html>

OGR classes, with new ones in red



Project: GDAL (2/2)

Student: Alexandr Borzykh

Mentor(s): Dmitry Baryshnikov, Even Rouault

Title: GDAL DWG support



Description: The aim of my project was to extend GDAL supported formats with DWG.

State of the project before GSoC: There was a DWG support, but it was not built-in by default, and GDAL Driver was based on third-party library Teigha (which is not X/MIT licensed, not even close).

Addition to the project: GDAL CAD Driver - new driver that uses libopencad as datasource. Libopencad is a library written in C++11, which provides a way to read/write CAD (DWG/DXF/DXFB) files. Most features that libopencad reads from a CAD file are mapped to OGR features.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003499.html>

Project: GRASS GIS (1/3)

Student: Adam Laža

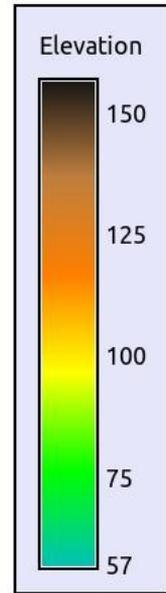
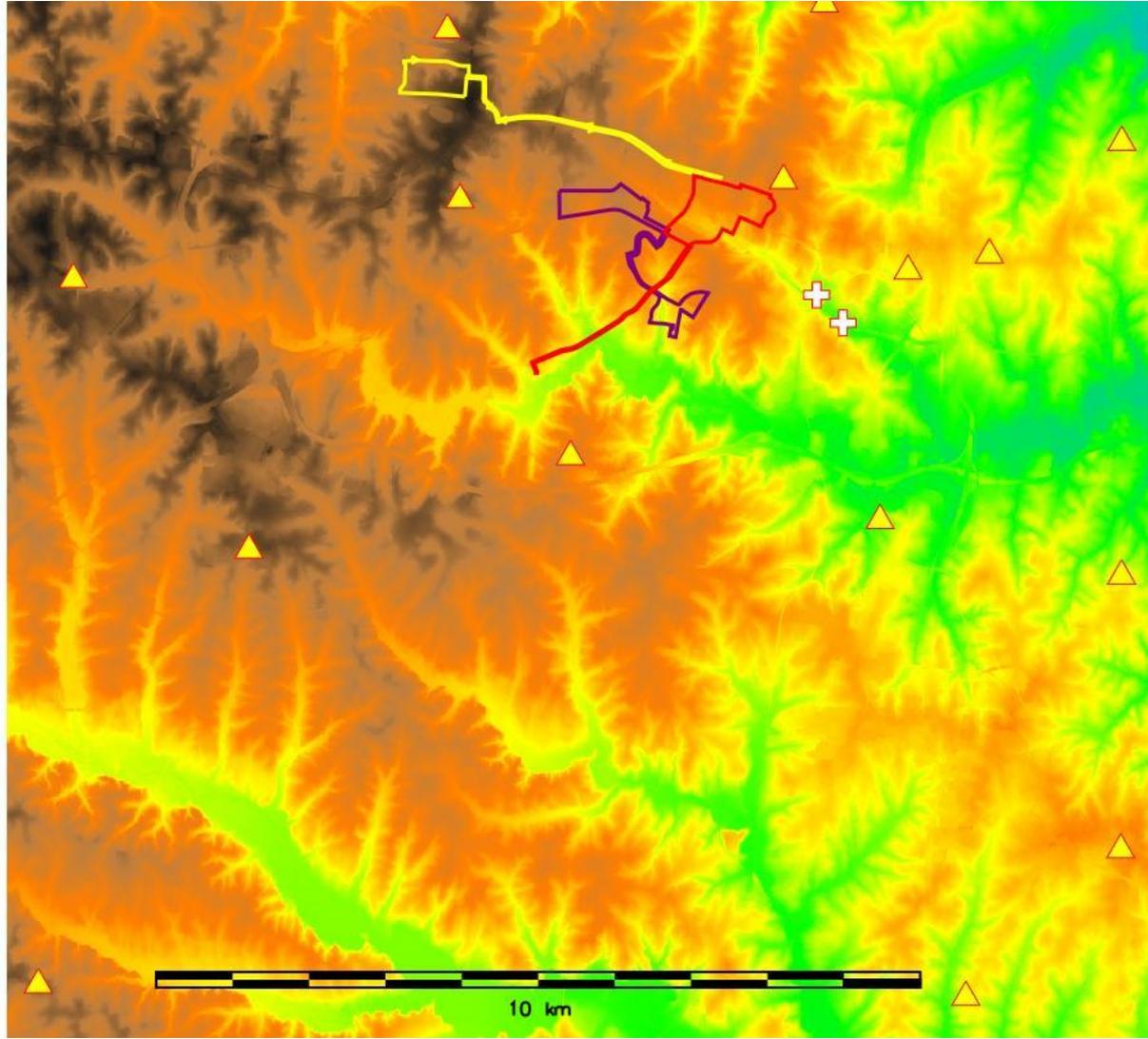
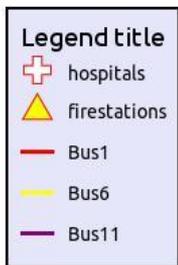
Mentors: Anna Petrasova, Vaclav Petras

Title: Complete basic cartography suite in GRASS GIS wxGUI Map Display



Description: The project aims to enhance some cartographic d.* modules, to fix some bugs in them and create a module for vector legend.

State of the project before GSoC: Before this summer some of the d.* modules were quite limited. User couldn't use full potential of modules and couldn't customize them, e.g. few symbols for north-arrow, option to rotate the arrow with/without label, set the length of scalebar, background settings... Before this summer there was no legend for vector maps.





Addition to the project:

d.histogram - enhance of displaying labels

d.grid - added options for labels background

d.northarrow - added several new symbols, added option for rotation and custom label, background settings

d.scalebar - added options for units, scalebar length, background settings

d.legend - support for border and background, added options for displaying ticks, added options for setting steps and labels, added logarithmic scale, enhanced histogram option, added option for units

workspace - enhance workspace setting behavior, include map elements into workspace

d.legend.vect - new module for displaying legend of vector maps, created new symbols for areas and lines

d.vect.thematic - enhance module for lines and points

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003480.html>

Project: GRASS GIS (2/3)

Student: Ondřej Pešek

Mentors: Vaclav Petras, Anna Petrasova



Title: GRASS GIS - PyQt implementation of GUI forms generated automatically from XML

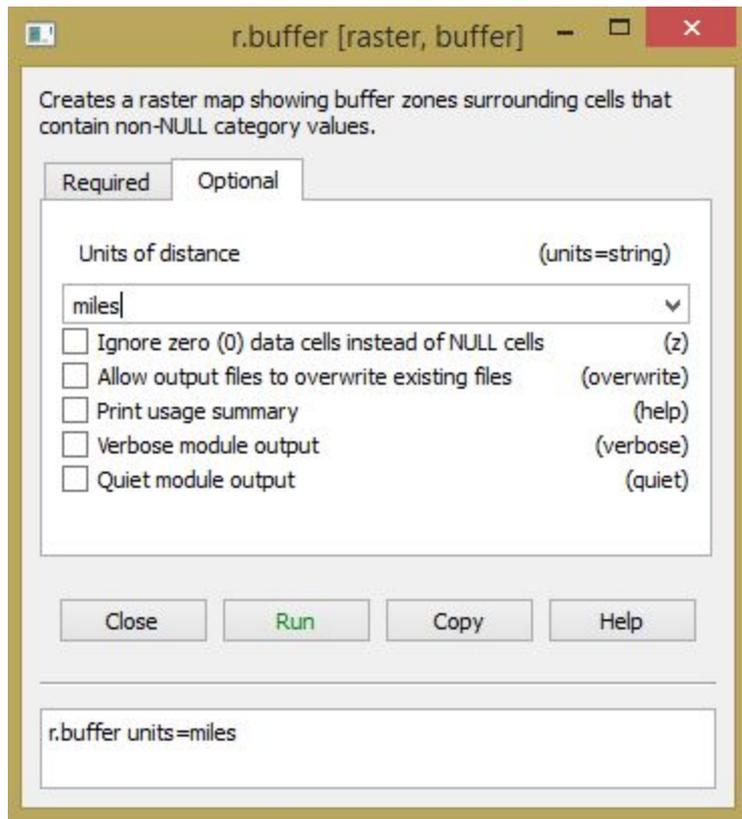
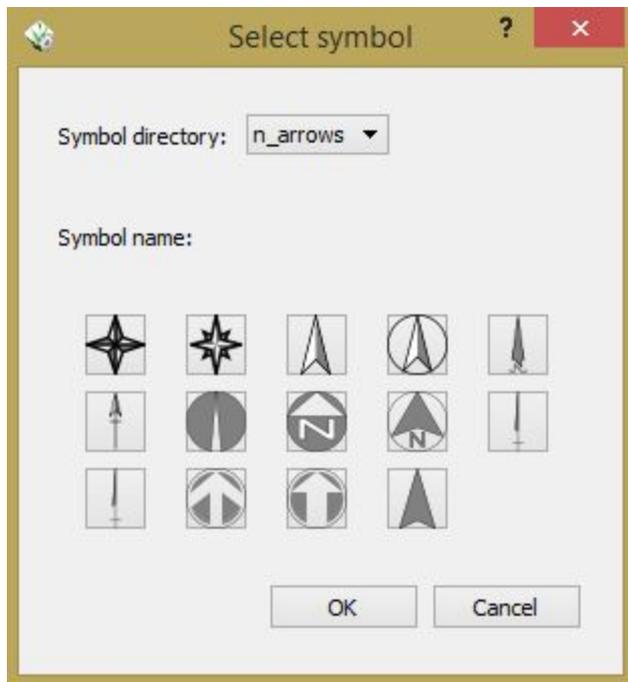
Description: The goal of the project was to create new GUI for modules based on PyQt.

State of the project before GSoC: The old one is based on wxPython. The reason is that there are some bugs in wxPython and PyQt is now much more powerful. The second reason is that the code of old GUI is badly written. It's almost impossible to read what something does and work with that. So the second goal was to try to write higher-quality code.

Addition to the project: Usable code that generates GUI forms with PyQt, with interfaces largely similar to WxPython ones, except for a few improvements.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003492.html>

New PyQt GUI forms:



Project: GRASS GIS (3/3)

Student: Mayank Agrawal

Mentors: Massimo di Stefano, Mohammed Rashad

Title: WEBGRASS

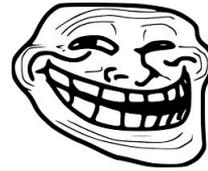
Description: My project focussed on the development of web based GUI for the GRASS “WebGrass” where people can use GRASS modules on their data through a web browser without actually installing GRASS.

GRASS is running on a server. The user interface is built using wt web toolkit.

State of the project before GSoC: not implemented, but a proof of concept was there.

Addition to the project: A working webapp. People can use data from their system and apply different modules and get output. A Web based UI like GRASS and modules implementation. Whole "WebGrass" is a very big project for one GSoC and we try to implement as much as possible.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003504.html>





grass gis

Bringing advanced geospatial technologies to the world

GRASS GIS DATA DIRECTORY

PROJECT LOCATION	PROJECT MAPSETS
<ul style="list-style-type: none">nc_basic_spm_grass7testLocationernewLocation	

Project: GvSIG Educa (1/2)

Student: Carlos I. Colombana

Mentors: Óscar Martínez, Joaquín del Cerro

Title: Add tests and educational games support to gvSIG Educa

Description: My project focused on the development of addons which allows gvSIG to create/play tests and educational games. The code was mainly developed in Jython, an implementation of the Python programming language written in Java, through the scripting module of the gvSIG desktop 2.x branch.

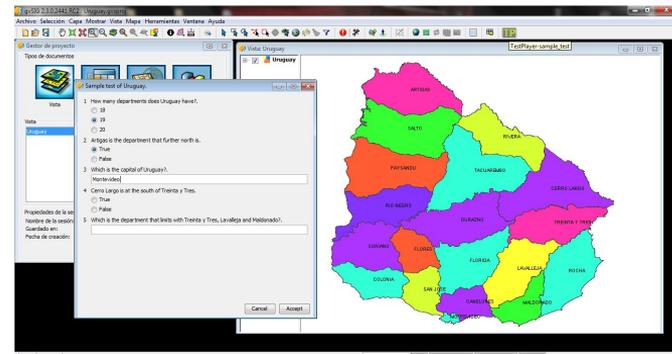
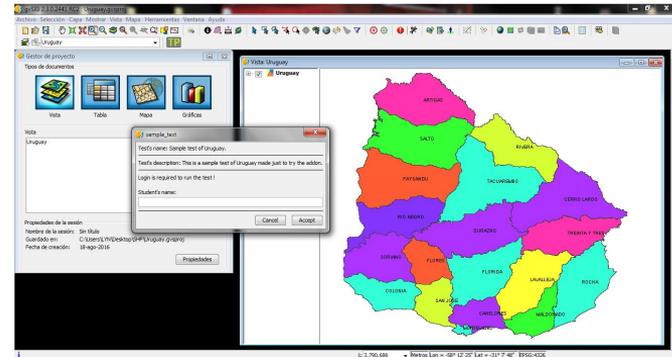
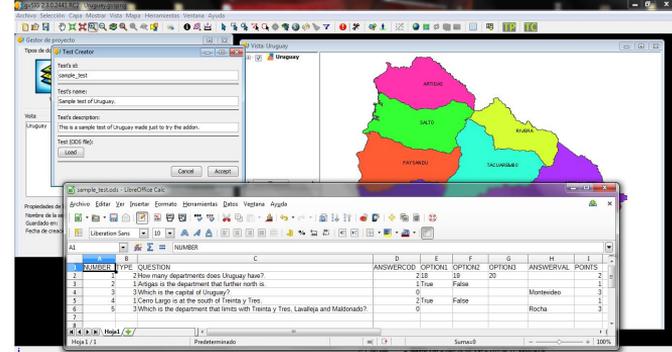
State of the project before GSoC: This functionality was missing.

Addition to the project: Now, with the "TestCreatorAndPlayer" addon, it's possible to create/play tests in gvSIG. Teachers can prepare the test easily in LibreOffice using the "sample_test.ods" file, load it and set test's main attributes in gvSIG, pack it with gvSIG built-in scripting framework and share it with students.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003521.html>



Teachers can prepare the test easily in LibreOffice using the "sample_test.ods" file, load it and set test's main attributes in gvSIG throw "TestCreator" ("TC") addon's app, pack it with gvSIG built-in scripting framework and finally share it with students. Students can install the test in gvSIG throw built-in addons manager, login and play it throw "Testplayer" ("TP") addon's app, and finally generate a report in pdf format. If answers and points were included by teachers, automatic correction is performed and feedback is given.



Project: GvSIG (2/2)

Student: Silvia Franceschi

Mentors: Andrea Antonello, Riccardo Rigon

Title: Development of a model for woody debris flooding hazard in gvSIG



Description: My project focused on the scientific improvement of the tools for the evaluation of Large Wood flooding in the JGrassTools library developed during the GSoC2014 and their porting in gvSIG.

State of the project before GSoC: the tools for the evaluation of the hazard derived from the transportation of Large Wood during floods used a simplified approach, that could be unrealistic.

Addition to the project: the tools for Large Wood Flooding are more completed and stable and it is possible to use them practically for the evaluation of the hazard derived from the transportation of woods in the river during flooding events without too many simplifications.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003508.html>



Project manager

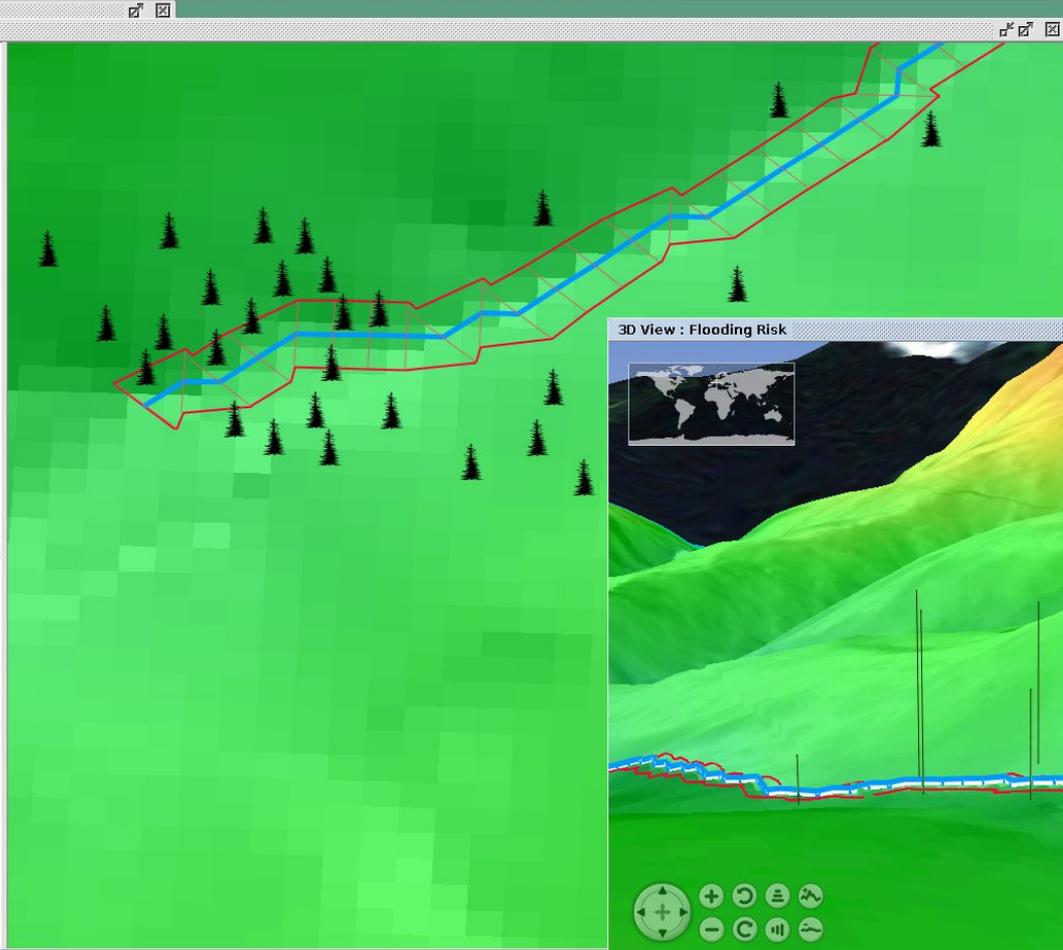
Document types:



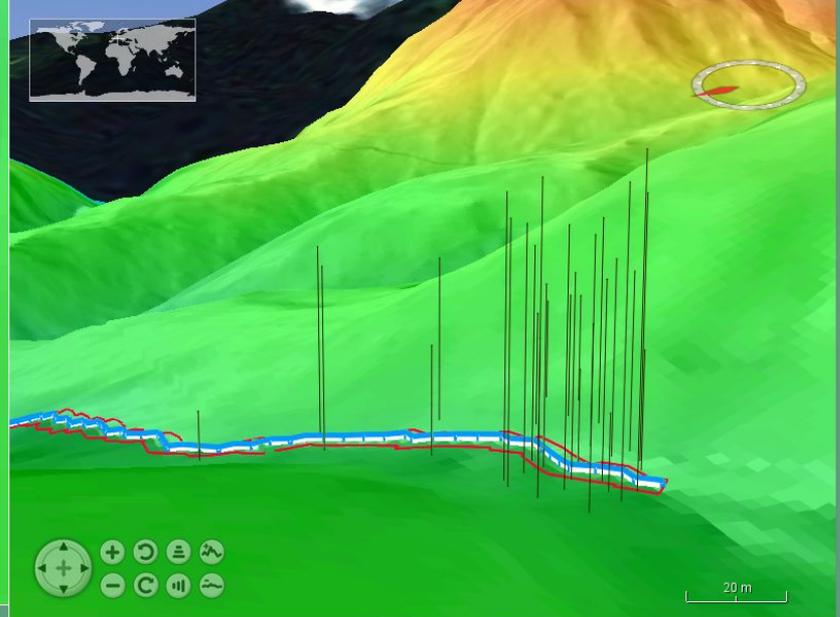
View: Flooding Risk

- tree_points
- extracted_ban
- extracted_net
- inund_area
- geology
- aspect
 - 0
 - 180
 - 360
- dtmfel
 - 1335
 - 1468
 - 1601
 - 1734
 - 1867
 - 2000

Session Name: U
 Saved as: /F
 Creation date: A



3D View : Flooding Risk



Altitude 2 km Off Globe

Project: istSOS (1/3)

Student: Florin-Daniel Cioloboc

Mentors: Mirko Cardoso, Milan Antonovic

Title: Android istSOS client



istSOS®

Description: My project focused on developing an Android Client for the istSOS community.

State of the project before GSoC: istSOS was only available as a platform that could be self-configured, developed in Python. My goal was to extend the possibilities of using istSOS in the Java-Android direction.

Addition to the project: I developed a core API Wrapper that exposes the communication of the core features of istSOS in Java, together with showcase demo-type that implements the app's core functionality.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003496.html>

istSOS Android client

Empowering Java developers with istSOS capabilities



*Open software for
standard sensor data*

Android demo

Shows...

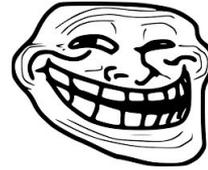
- istSOS in Android environment
- How to develop customized Android istSOS apps

istSOS Java Core

Developed as...



- Library for developing Java apps that need istSOS
- Has IstSOS core functionalities implemented
- Suitable for Android development



Project: istSOS (2/3)

Student: Luka Glusica

Mentors: Massimiliano Cannata, Milan Antonovic

Title: istSOS web API

Description: implementation of an easy to use framework that will offer to the final user the ability of building feature-rich web widget that will be easily embeddable into existing web sites.

State of the project before GSoC: IstSOS data couldn't be published on the Web, without hard coding from scratch. Developers would have had to learn about WA REST interface.

Addition to the project: With istSOS Web API, developers can use powerful istSOS core JavaScript library that covers most of the WA REST HTTP requests. Also, developers now have istSOS Web widget creator, at their disposal: an application that uses istSOS core Javascript library and offers a set of widget creator tools.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003498.html>

MAP TOOL CHART TOOL BOX TOOL

ISTSOS WEB WIDGET CREATOR

istSOS Web Widget Creator

An Online Web Application for building featurerich web widget that will be easily embeddable into existing web pages.

istSOS core library

IstSOS Web widget creator uses istSOS core JavaScript library for interacting with the istSOS server and its data.

Use tools or customize even more

Developers can use Web widget creator tools to easily generate embeddable code, but, considering the fact that this application is built upon istSOS core JavaScript library, developers can also use the library directly to raise customization level even more

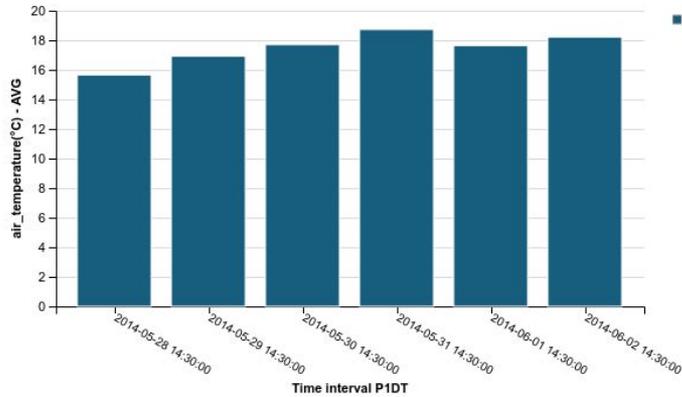
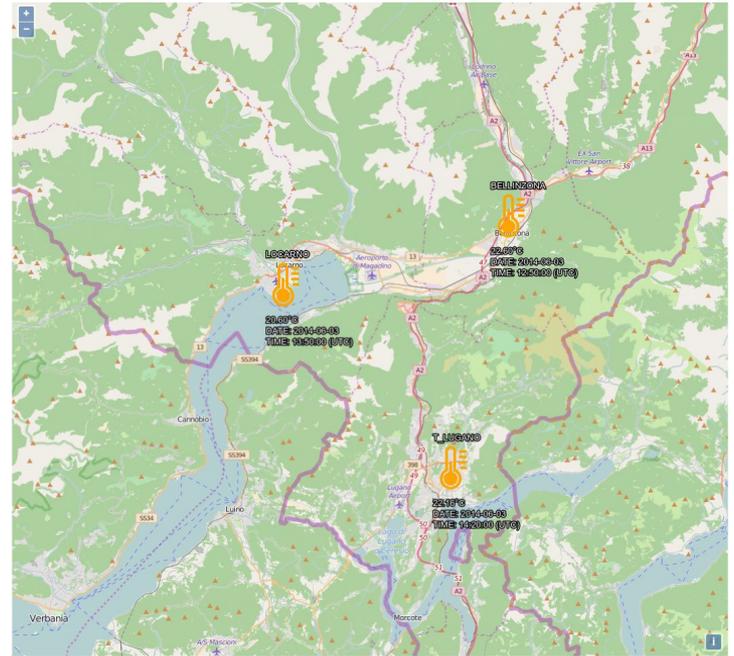
Links

- [istSOS - OGC SOS server implementation](#)
- [istSOS core JavaScript library](#)
- [istSOS Web widget creator](#)

Links

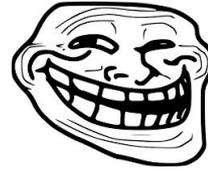
- [istSOS - OGC SOS server implementation](#)
- [istSOS core JavaScript library](#)
- [istSOS Web widget creator](#)

Embedded code:



Export as PNGView SourceOpen in Vega Editor

PROCEDURE: GRABOW	DATE: 2014-05-27	TIME: 22:00:00
	SOLAR RAD.: 0.00 W/m²	RAINFALL: 0.00 mm
	WIND: 5.00 m/s	HUMIDITY: 92.20 %
	TEMPERATURE: 9.00 °C	



Project: istSOS (3/3)

Student: Felipe Poveda

Mentors: Milan Antonovic, Massimiliano Cannata

Title: VistSOS: the istSOS Data Visualization Framework

Description: the development of a framework for defining chart visualizations of sensor observations managed by istSOS.

State of the project before GSoC: istSOS didn't have a tool to create charts of observations in a Web environment. Therefore, the implementation started from zero, exploring technologies and approaches to achieve the goals and taking design decisions to fix the issues that appeared along the way.

Addition to the project: capability of defining charts for time series and aggregation statistics on the observations managed by the system istSOS.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003497.html>



Project: NASA World Wind

Student: Gabriele Prestifilippo

Mentors: Jakub Balhar, Patrick Hogan

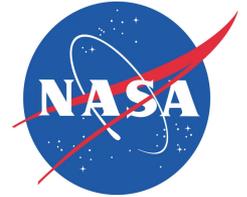
Title: Multidimensional Visualization Tool for Environmental Variables

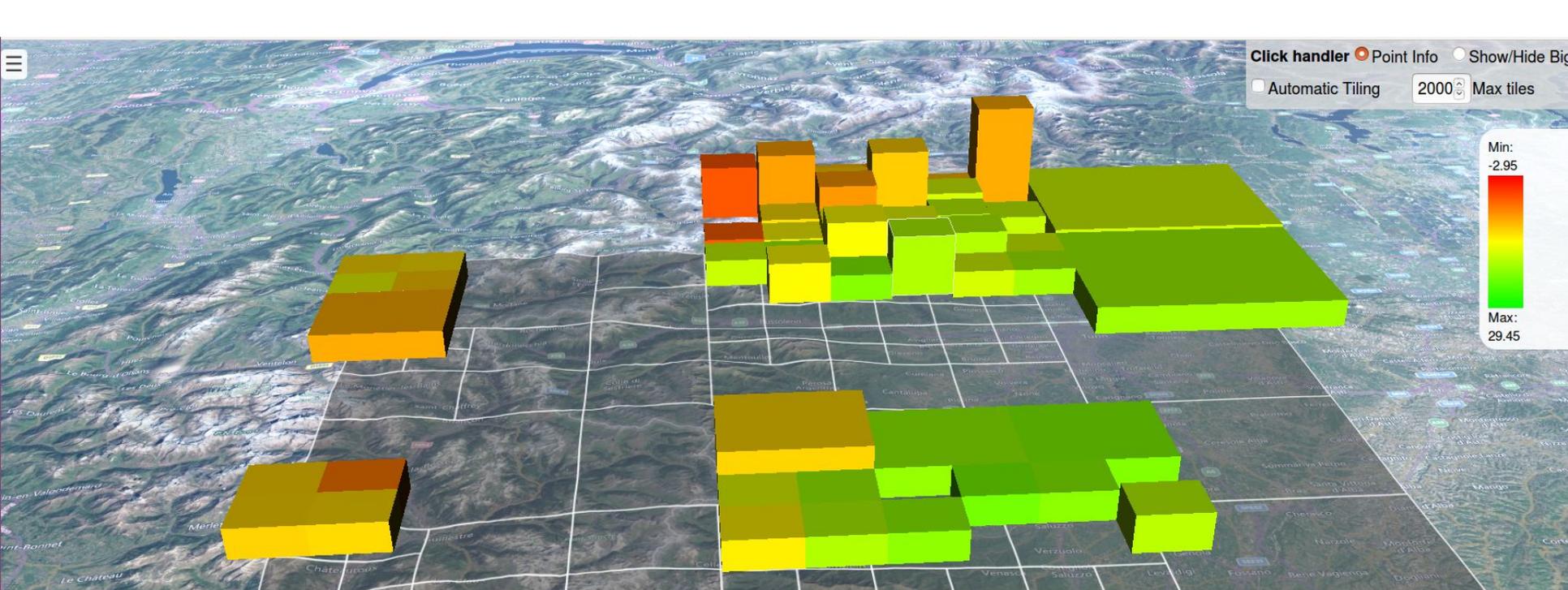
Description: development of a Web Application, using the NASA Web WorldWind framework, to create a tool capable of visualize multidimensional datasets in more than three dimensions.

State of the project before GSoC: There was no similar project before developed with NASA Web WorldWind.

Addition to the project: application for visualising multidimensional datasets in a 3D environment, using different models.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003491.html>





Correlation between variables: 0.8430497

Correlation between datasets in selected voxel: -0.26041

Hide Statistics



Project: OpenLayers 3

Student: Samuel Lapointe

Mentors: Alexandre Dube, Jessica Lapointe

Title: OGC protocols support within OL3-Google-Maps



Description: add support for new types of layers in the OL3-Google-Maps library. This meant that the layers using the WMS, XYZ and TMS protocols would be rendered using the Google Maps API.

State of the project before GSoC: these layers would be rendered using OpenLayers 3 and placed above the Google Maps layer. There was a visible desynchronisation when panning and zooming in the map, as OL3 and Google Maps don't animate these actions the same way.

Addition to the project: most tile protocols are supported in the library, as well as the ImageWMS layer type. These layers are rendered by Google Maps itself, so there's no possible desynchronisation.

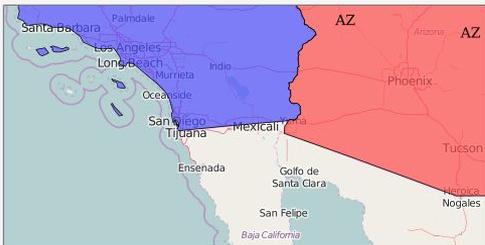
Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003479.html>

OpenLayers 3

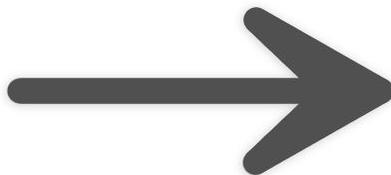
The user adds layers and features using the OpenLayers 3 syntax:

```
var tileWMSLayer = new ol.layer.Tile({
  source: new ol.source.TileWMS({
    ...
  }),
  visible: true
});

var map = new ol.Map({
  ...
});
```



ol3gm



Google Maps

OL3-Google-Maps translate the calls to functions on the Google Maps API

```
var options = {
  ...
};

var googleTileLayer =
  new google.maps.ImageMapType(options);

this.gmap.overlayMapTypes.push(
  googleTileLayer
);
```



Project: OneBusAway - transiTime

Student: Brendan Egan

Mentors: Og Crudden, Stefan Steiniger, Sean Barbeau

Title: transiTime QuickStart Application

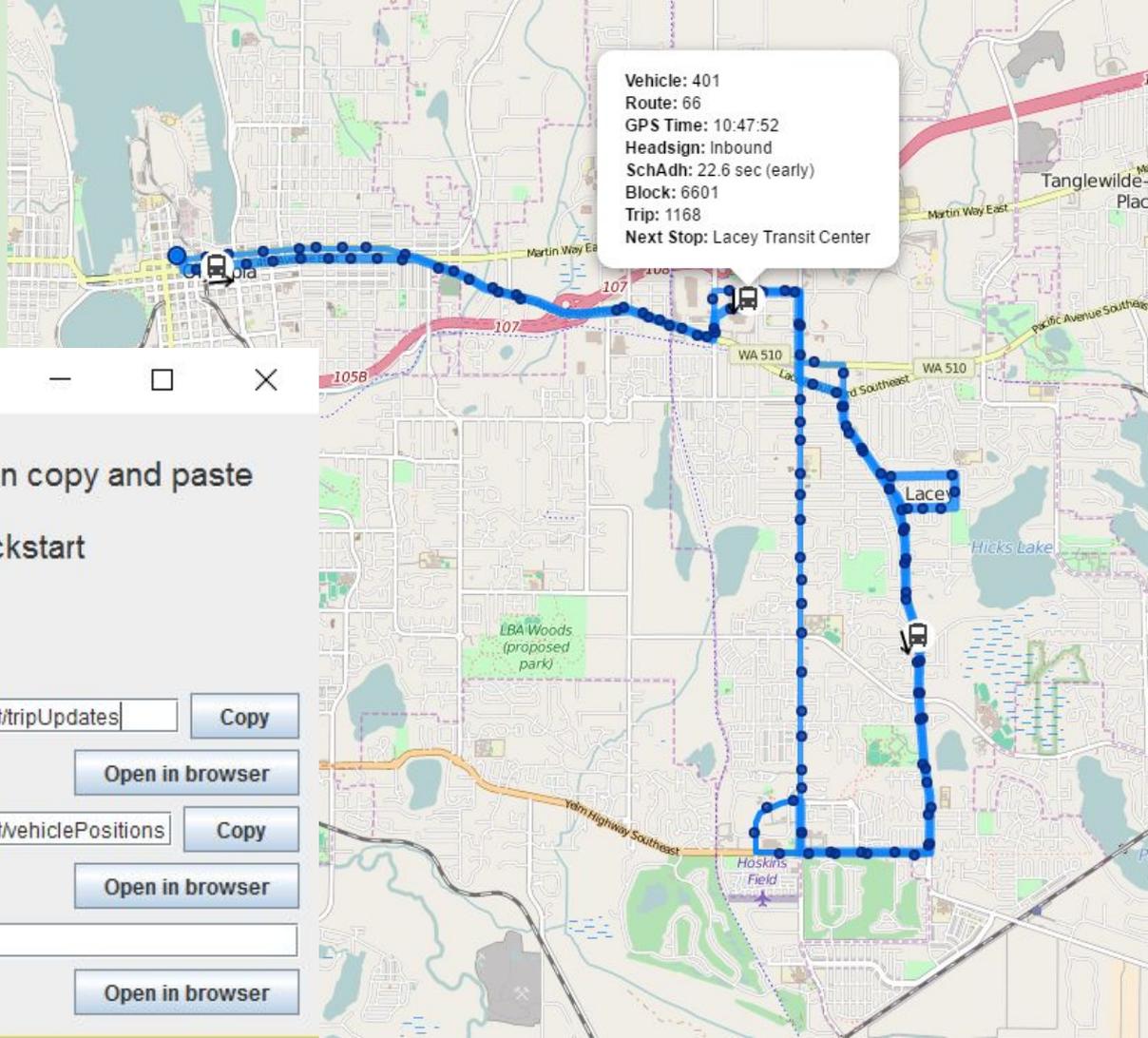


Description: create a QuickStart for the Transitime Application, transiTime is a real-time transit information system which provides provide real-time arrival information such as "next bus is in 5 minutes". It does this by taking a GTFS static file and vehicle location feed and using those produces predictions for stops in the form of GTFS trip updates.

State of the project before GSoC: the setup process of transiTime was very difficult with little available documentation.

Addition to the project: working installation running from an executable jar within a few minutes.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003514.html>



Vehicle: 401
Route: 66
GPS Time: 10:47:52
Headsign: Inbound
SchAdh: 22.6 sec (early)
Block: 6601
Trip: 1168
Next Stop: Lacey Transit Center

transiTimeQuickStart

Installation is now up and running! you can copy and paste these links into the OneBusawayQuickstart

Trip Updates URL

<http://127.0.0.1:8080/api/v1/key/65562734/agency/01/command/gtfs-rt/tripUpdates>

Copy

Vehicle Positions URL

Open in browser

<http://127.0.0.1:8080/api/v1/key/65562734/agency/01/command/gtfs-rt/vehiclePositions>

Copy

transitime Webapp address

Open in browser

<http://127.0.0.1:8080/web>

Open in browser



Project: OSSIM

Student: Martina Di Rita

Mentors: Oscar Kramer, Dave Burken

Title: A complete photogrammetric OSSIM tool for automatic DSMs generation using multi-view optical and SAR images

Description: improvement and the implementation of new functionalities for DATE (Digital Automatic Terrain Extractor) OSSIM plug-in.

State of the project before GSoC: DATE was an OSSIM plug-in for DSMs generation from optical stereo-pair and triplets, not able to process SAR images; the pyramidal approach implemented was not automatic.

Addition to the project: now it is possible to generate automatically DSMs from SAR imagery using an iterative coarse-to-fine approach, to take advantage of iterative solutions at gradually increasing resolution.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003516.html>

A PHOTOGRAMMETRIC OSSIM TOOL

Main features added:

- Automatization of the pyramid coarse-to-fine approach
- Implementation of SAR imagery stacks processing
- DSMs merging

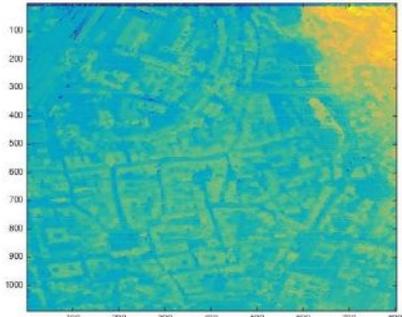


Fig.1

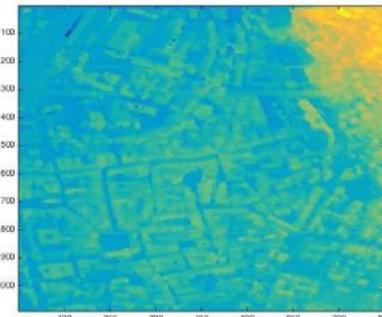


Fig.2

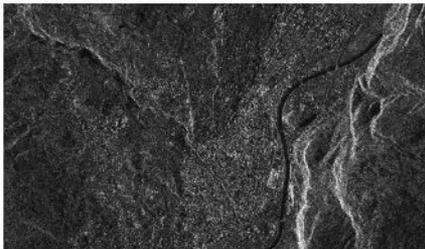


Fig.3

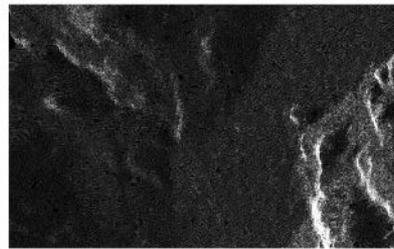


Fig.4

DSMs merging strategies:

1) regularization through edge information: Fig.2 – the regularized one – shows far sharper edge than Fig.1 – showing the mean)

2) automatic mask generation for areas with geometric distortion due to foreshortening and layover effects. The black areas (mask in Fig.4) correspond to those with less distortions (SAR img in Fig.3) whereas the white ones are those that show the greatest geometric distortions

Project: pgRouting (1/2)

Student: Andrea Nardelli

Mentors: Daniel Kastl, Vicky Vergara

Title: Flow Algorithms for pgRouting



Description: adding functionality to solve maximum flow problems to pgRouting library.

State of the project before GSoC: missing functionality.

Addition to the project: Implementation of 3 algorithms and related applications:

- Edge disjoint paths
- Maximum cardinality matching
- Multiple sources/sink flow

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003518.html>

pgRouting Maximum Flow

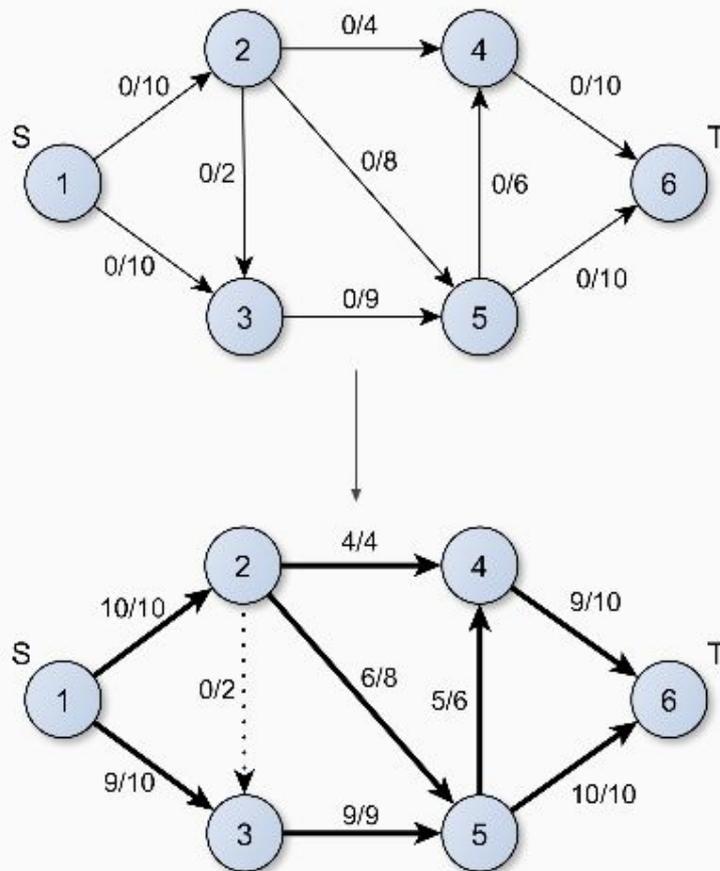
Maximum flow algorithms route the maximum amount of one commodity from one source S to one sink T.

This functionality was added to the library, in addition to some example applications:

- Edge disjoint paths
- Multiple source/sink flow
- Maximum cardinality matching

Code of example:

https://github.com/lledran/pgrouting/blob/gs-oc-flow/src/max_flow/demo/slide_demo.sql#L22



Project: pgRouting (2/2)

Student: Sankepally Rohith Reddy

Mentors: Vicky Vergara, Daniel Kastl

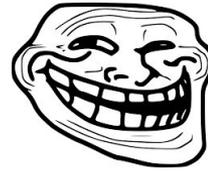
Title: Implementation of a framework which supports addition of contraction techniques for pgRouting

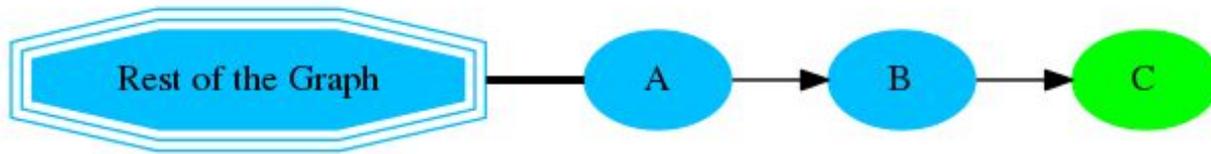
Description: In big graphs, like the road graphs, or electric networks, graph contraction can be used to speed up some graph algorithms. Contraction reduces the size of the graph by removing or modifying some of the vertices and edges, thus decreasing the total time and space used in graph algorithms.

State of the project before GSoC: the functionality was missing.

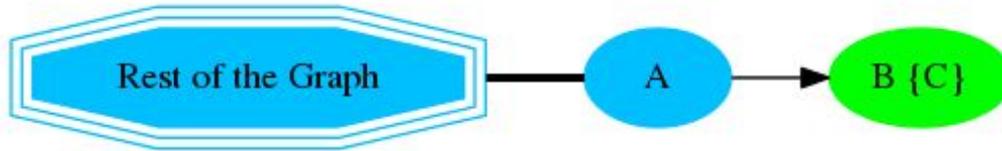
Addition to the project: Implementation of dead end and linear contraction, with options to preserve original graph sections and control the number of algorithm cycles.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003519.html>





After contracting c , node B is now a *Dead End* node and is contracted:

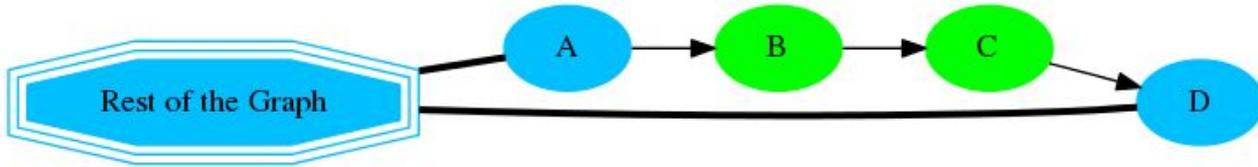


Node B gets contracted

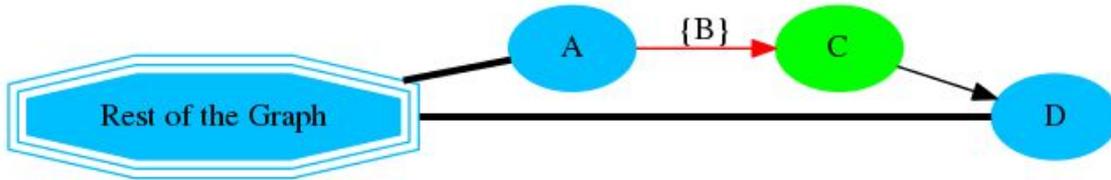


Nodes B and C belong to node A .

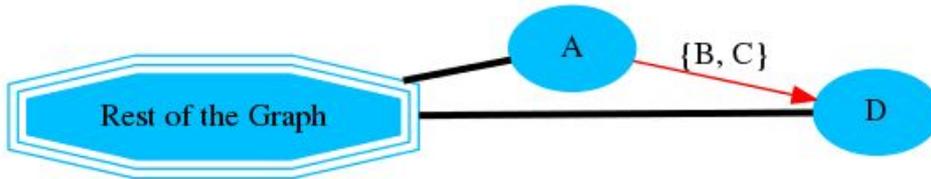
Dead end contraction



After contracting B, a new edge gets inserted between A and C which is represented by red color.



Node C is *linear node* and gets contracted.



Nodes B and C belong to edge connecting A and D which is represented by red color.

Linear contraction

Project: pyWPS

Student: Jan Rudolf

Mentors: Jonas Eberle, Jachym Cepicky

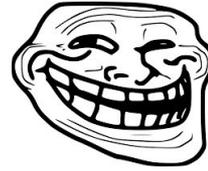
Title: Web-based administration & process management for PyWPS

Description: extend PyWPS with a way to control processes and provide a web interface for it.

State of the project before GSoC: there was no way to manipulate running processes. There also wasn't any web interface for this purpose.

Addition to the project: On the shoulders of Flask framework, I've built the REST API for manipulating processes and the web interface. The user can pause, resume and stop processes via the web interface. The database connection was changed from hard-wired to SQLAlchemy ORM.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003523.html>



PyWPS



Manage Processes

[x clear filter](#)

Status:

Operation:

Identifier:

UUID:

PID:

Action	Status	UUID	PID	Operation	Identifier	Message	Percent Done
Pause Stop	Running	ccbab459-67ce-11e6-91ff-58b035fc9e02	11714	execute	sleep	Process running...	20.0

Project: QGIS

Student: Akbar Gumbira

Mentors: Alessandro Pasotti, Anita Graser

Title: QGIS Styles, Symbols, and SVG Markers Sharing Repository

Description: The project idea is to make svg, symbol, and style sharing easier. The idea of the implementation is changed (from web service architecture to distributed directory based repository system).

State of the project before GSoC: There was no tool that offered this service..

Addition to the project: With this new plugin it's easier to share resources among QGIS users and hopefully we will have rich collections (whether it's icons, svg, images) to make cool maps.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003493.html>



QGIS Resource Sharing

All
Installed
Settings

Search

- Cultuur Historische Waar...
- Flowmap
- Mapbox's Maki Icons**
- North Arrow
- OSM Spatialite Googlemaps
- OpenStreetMap style
- Road Signs

Mapbox's Maki Icons

Maki is an icon set made for map designers created by Mapbox

URL <https://github.com/anitagraser/QGIS-style-repo-dummy.git>

Tags maki, icon, svg, mapbox

Author Akbar Gumbira

E-mail akbargumbira@gmail.com

Open folder Install Uninstall

Help Close

Project: ZOO-Project (1/2)

Student: Chingchai Humhong

Mentors: Luca Delucchi, Gérald Fenoy

Title: Bringing pyModis to the web through ZOO-Project

Description: The pyModis project has been developed and used to work with MODIS data, it provides wxPython user interfaces which are able to download and process data using pyModis scripts.

State of the project before GSoC: pyModis was able to run on a local computer but cannot be directly remotely invoked on-demand. ZOO-Project was able to handle services implemented in the Python language but do not offer any pyModis capabilities.

Addition to the project: By providing pyModis as a service through ZOO-Project, capabilities of ZOO-Project will be increased, specially capabilities of python in ZOO-Project will be exposed.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003520.html>



Modis Services parameters

mosaic modis data from hdf to GDAL formats using GDAL.

DataInputs

tiles the directory where the data that you want to download are stored on the FTP server.

ProcessOutputs

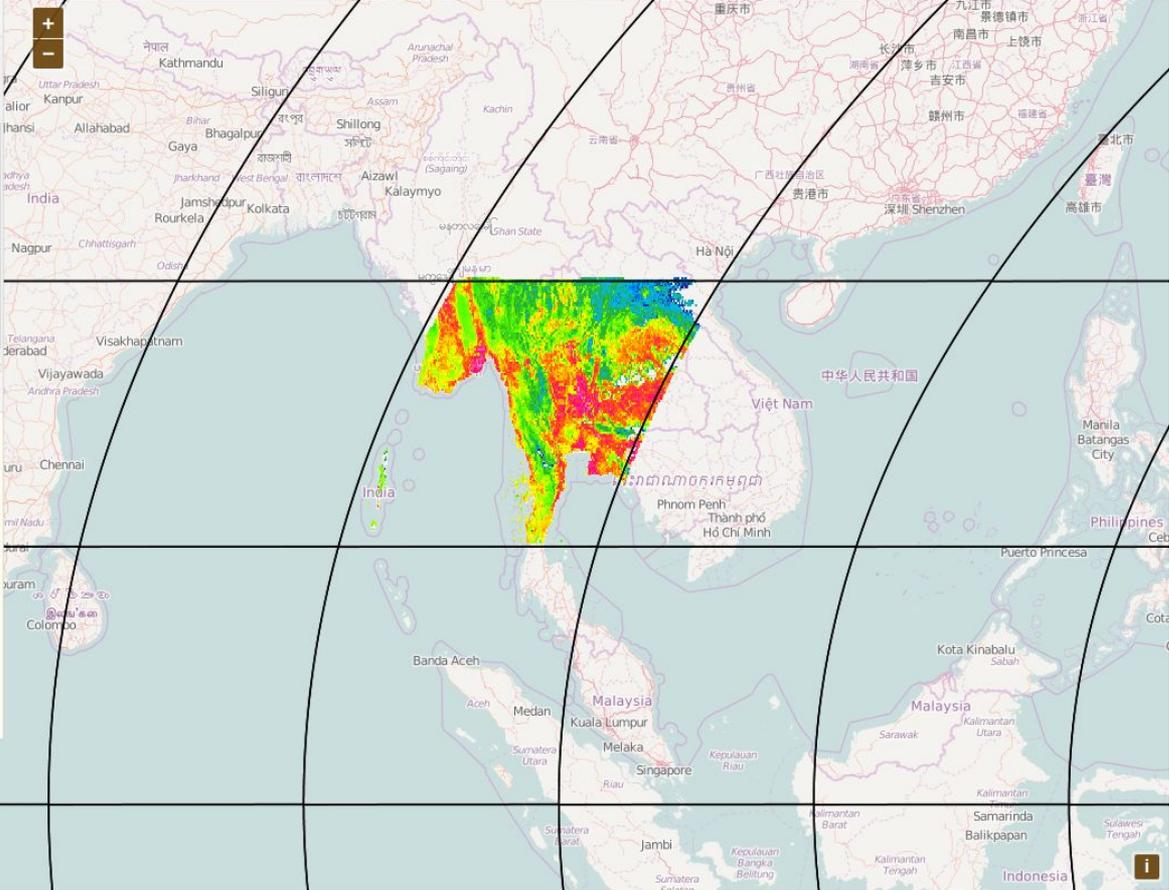
Result output modis_mosaic.py

image/tiff

Execute

Layers

- Modis Layer



Project: ZOO-Project (2/2)

Student: Niroshan Sanjaya

Mentors: Gérald Fenoy, Andrea Antonello



Title: Implementing WPS for Geoparazzi field data collection tool using ZOO-Project: Simplifying integration of field data and GIS

Description: This project mainly focus on enhancing the capabilities of ZOO-Project by providing JGrasstools functions as services. Secondly, focus on implementing web-based platform for conversion of data from Geoparazzi and provide data as WFS for further use.

State of the project before GSoC: ZOO-Project, JGrasstools and Geoparazzi are separate tools.

Addition to the project: We have successfully integrated Geotools with ZOO-Project. Through that any type of Java functions which is available on Geotools can be implemented.

Link: <https://lists.osgeo.org/pipermail/soc/2016-August/003517.html>

Too much information at once?





Google
Summer of Code



Find more info and links on OSGeo Wiki:

https://wiki.osgeo.org/wiki/Google_Summer_of_Code_2016_Results