

Results of Google Summer of Code 2015 at

OSGeo



Margherita Di Leo

Anne Ghisla

GSoC 2015 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**. Since its inception in 2005, the program has brought together over 8,500 successful student participants from over 101 countries and over 8,300 mentors from 109 countries worldwide to produce over 55 million lines of code. “

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

What is  OSGeo ?

OSGeo

The Open Source Geospatial Foundation

Visit: osgeo.org

2016

2005

Need to reorganize 200+ FOSS4G projects

Feb 2006

OSGeo founded by GDAL, GeoTools, GRASS, MapBender, MapBuilder, MapGuide Open Source (Autodesk), Mapserver, OSSIM

OSGeo Projects

Web Mapping

deegree
geomajas
GeoMOOSE
GeoServer
Mapbender
MapBuilder
MapFish
MapGuide Open Source
MapServer
OpenLayers

Desktop Applications

GRASS GIS
gvSIG
Marble
QGIS

Geospatial Libraries

FDO
GDAL/OGR
GEOS
GeoTools
OSSIM
PostGIS

Metadata Catalogs

GeoNetwork
pycsw

Outreach Projects

Public Geospatial Data
Education and Curriculum
OSGeo Live

Incubating Projects

Web Mapping

istSOS
PyWPS
Team Engine
ZOO-Project

Desktop Applications

Opticks

Geospatial Libraries

MetaCRS
Orfeo ToolBox (OTB)
rasdaman

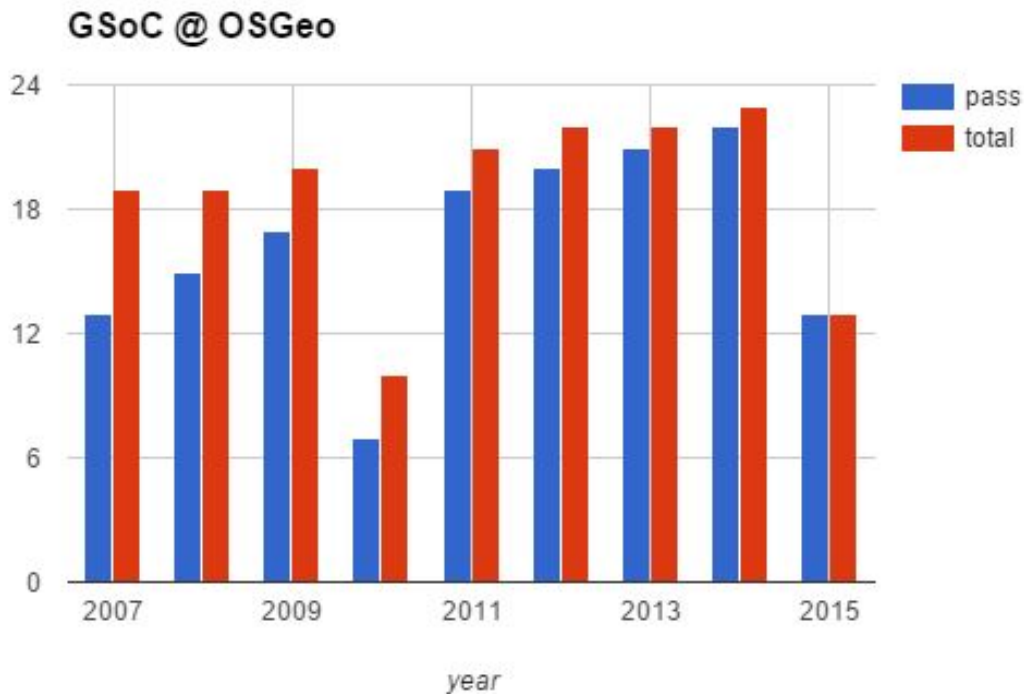
Content Management Systems

GeoNode

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



More at http://wiki.osgeo.org/wiki/Google_Summer_of_Code_Application_2015

GSoC 2015 @ OSGeo

13 students accepted

Project: GDAL

Student: Faza Mahamood

Mentor(s): Even Rouault

Title: Integration of GDAL utilities into GDAL core library



Description: Calling GDAL utilities from code without involving system calls, to be able to work on in-memory datasets and use progress/cancellation callback functions.

State of the project before GSoC: in order to use any functionalities of GDAL utilities in a program, the developer had to use system calls. Even then, the developer cannot work with in memory datasets.

Addition to the project: now, the developers can easily have all the functionalities of GDAL utilities through dedicated functions which are part of GDAL core library. There is no system call involved and it works on in-memory datasets.

Link: https://trac.osgeo.org/gdal/wiki/rfc59.1_utilities_as_a_library

Project: GRASS GIS

Student: Matej Krejci

Mentor(s): Martin Landa, Angelos Tzotsos, Luca Delucchi

Title: Improved Metadata for GRASS GIS

Description: The first aim was to improve current package by adding a new functionality for extend metadata handling. The second goal was to implement module for interaction with metadata catalogue services (CSW).

State of the project before GSoC: In 2014 during GSoC I have developed new GRASS GIS metadata management support based on ISO-INSPIRE. This package needed extension to meet the users requirements. In addition package for CSW searching was missing.

Addition to the project: New module g.gui.cswbrowser based on graphical interface allows managing csw connections, setting filter and searching, browsing results and adding services to GRASS GIS have been implemented. For g.gui. metadata have been added support: exporting metadata reports to pdf, storing themes of metadata in SQL backend, publishing metadata to CSW and metadata support for Temporal datasets.

Links: Project: https://grasswiki.osgeo.org/wiki/ISO/INSPIRE_Metadata_Support

Development page 2015: <https://trac.osgeo.org/grass/wiki/GSoC/2015/ImprovedMetadata>

source SVN: <http://svn.osgeo.org/grass/grass-addons/grass7/gui/wxpython/wx.metadata/>



Find **Configure**

Search settings

Catalog

Nationaal Georegister (Nederland)

Filter

All

Keywords

☐ Advanced

Extent

Xmax

180

Ymax

90

Xmin

-180

Ymin

-90

Global

Map extent

Max 20

Find

Showing 1 - 20 of 8861 result(s)

Type	Service
dataset	Schiphol IMU-1 externe veiligheid 1999
dataset	Schiphol IMU-1 externe veiligheid 2005
dataset	Schiphol NA60 geluidsniveau's 2004
dataset	Schiphol Lden geluidscontouren 1997
dataset	Schiphol IMU-3 externe veiligheid 2001
dataset	Schiphol NA70 geluidsniveau's 1994

<<

<

>

>>

Request XML

Response XML

Add WMS

Add WFS

Add WCS

Record Metadata ([View XML](#))

Identifier {BA2AC79D-E0DB-4195-9536-1FB02A110FED}

Title Schiphol Lden geluidscontouren 1997

Geluidscontouren 1997, uitgedrukt in Lden. De Lden (Loudness day-evening-night) is de Europese dosismaat voor de geluidsbelasting van verkeer en industrie welke is geïntroduceerd in 2003. Hierbij wordt niet langer gewerkt met zones en zoneringpunten, zoals gebruikt bij de Ke, maar met zogenaamde handhavingspunten. In elk van de handhavingspunten, die zijn gelegen in de buurt van de bebouwing rondom Schiphol, is een maximale geluidsbelasting vastgesteld die niet mag worden overschreden. De geluidsbelasting wordt gemeten in tijdsblokken binnen een etmaal; day van 7u tot 19u, evening van 19u tot 23u en night van 23u tot 7u. De geluidsbelastingen tijdens de avond en tijdens de nacht worden respectievelijk met een factor 5 en een factor 10 zwaarder geteld dan de geluidsbelasting overdag. De meting van de geluidsbelasting beslaat een geheel jaar en wordt uitgedrukt in dB(A), d.w.z. een decibelmeting met A-filter. Met het A-filter

Abstract

Project: gvSIG

Student: Eva Rodriguez

Mentor(s): Joaquín José del Cerro, Fran Peñarrubia

Title: Port Network Analysis Extension to gvSIG 2.x branch



Description: The aim was to develop a plugin to migrate all features of route calculation from the 1.* gvSIG version to the latest one. These are the main tasks: - Route management to get the shortest path - Optimizing the order of stops on a route - Calculation of service areas - Location of the nearest event - Calculate matrices Origin - Destination - Calculation of minimum spanning tree

State of the project before GSoC: All of this functionality does not exist into 2.* gvSIG version.

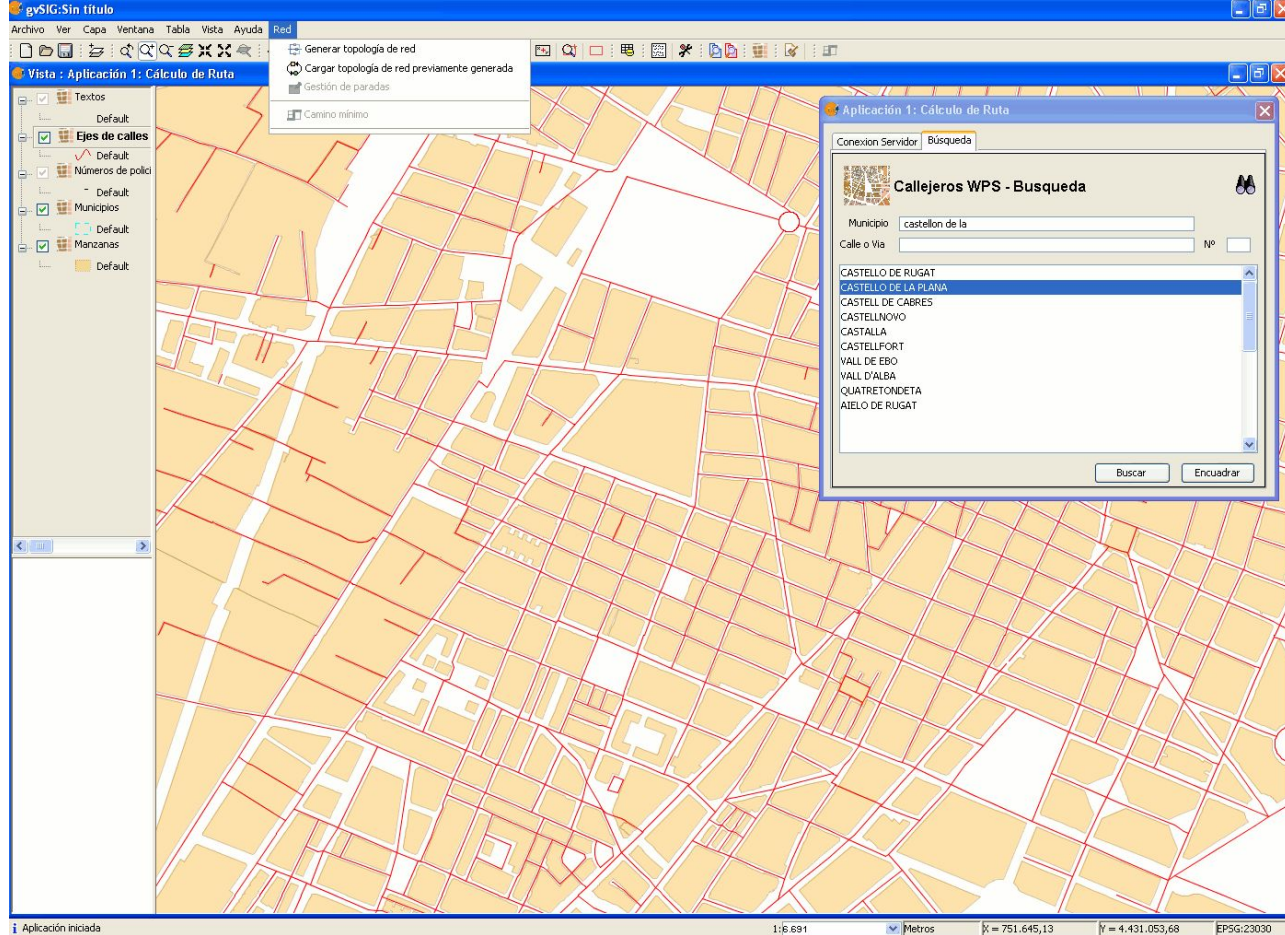
Now, users can use some functionality that existed in the 1.* gvSIG version.

Links: Installers: <https://redmine.gvsig.net/redmine/projects/gvsig-networkanalysis/files>

Source SVN: <http://devel.gvsig.org/svn/gvsig-networkanalysis/>

Documentation network plugin: <http://docs.gvsig.org/plone/docdev/docs/desarrollo/plugins/redes/>

Use case: <http://docs.gvsig.org/plone/docusr/acceso-editores/funcionalidades/servicio-de-callejeros/aplicaciones-tutoriales/CASO-2-calculo-de-rutas/?searchterm=redes>



From use case:

<http://docs.gvsig.org/plone/docusr/acceso-editores/funcionalidades/servicio-de-callejeros/aplicaciones-tutoriales/CASO-2-calculo-de-rutas/?searchterm=redes>

Project: istSOS

Student: Luca Ambrosini

Mentor(s): Massimiliano Cannata, Milan Antonovic

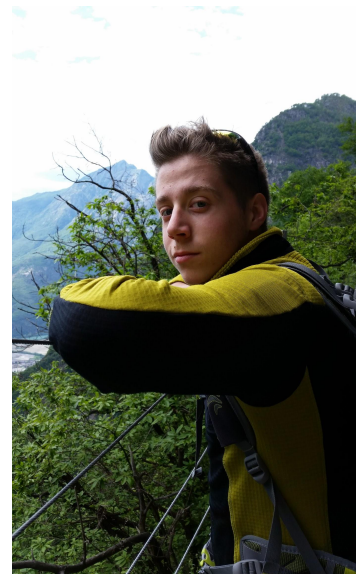
Title: Scalability for Big data processing for istSOS

Description: to implement data replication to improve security, online time and performance.

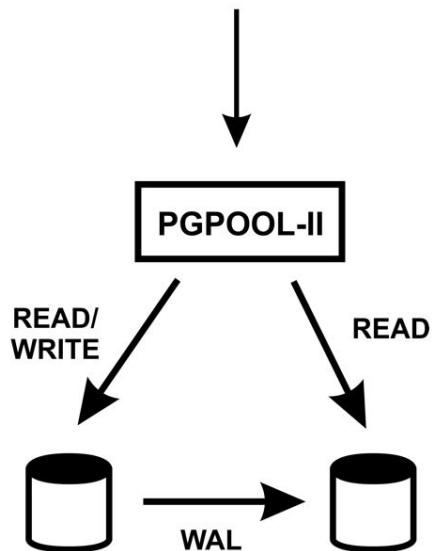
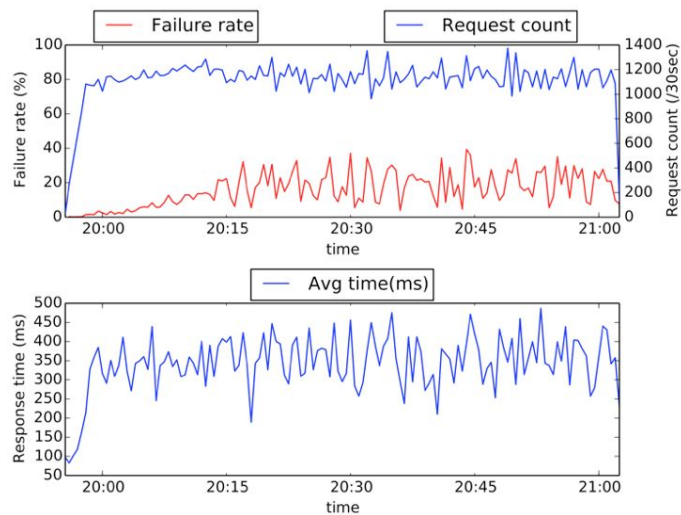
State of the project before GSoC: All data were hosted on a single postgres DB, and there wasn't a tool that helps finding this implementation limits.

Addition to the project: We adapted the source code to work with replications, we set up a middleware that balances query among multiple database server, and we realized a benchmarking tool to test system response time and its scaling .

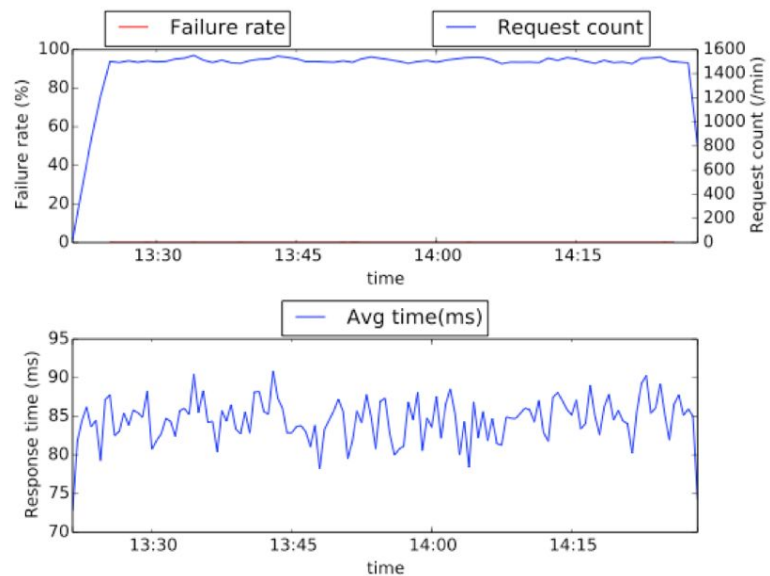
Links: Gsoc Reports : https://sourceforge.net/p/istsos/wiki/GSoC_2015_scalability/



Before



After



Project: JGrassTools

Student: Silvia Franceschi

Mentor(s): Andrea Antonello, Riccardo Rigon

Title: Development of a simplified 1D hydraulics model



Description: The module developed during this GSoC2015 is SaintGeo, a simple 1D hydraulic model based on the equations of Saint Venant.

State of the project before GSoC: There was an original implementation of the model in C language done during a master thesis by Angelo Zacchia and Riccardo Rigon. I did the porting in the JGrassTools library of the model together with the implementation of the code for reading and writing GIS layers as input and outputs.

Addition to the project: The new model bases on GIS input layers instead of text files and has a graphical interface, directly provided by the integration in the JGrassTools library to specify the inputs and outputs files.

Links: Project: <http://www.jgrasstools.org>

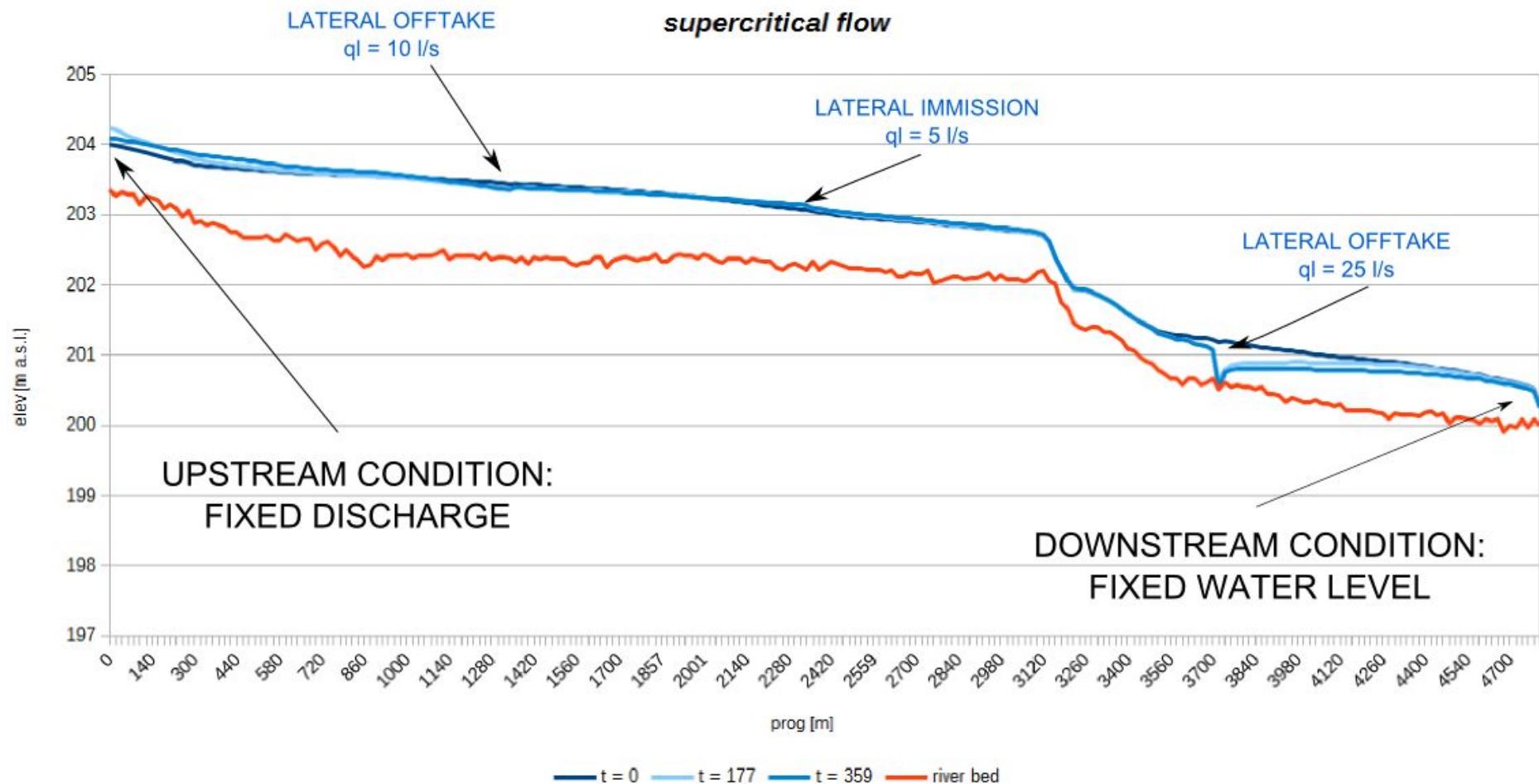
Development page 2015: <https://github.com/moovida/jgrasstools/wiki/Google-Summer-of-Code-2015>

Source: <https://github.com/moovida/jgrasstools/tree/master/hortonmachine/src/main/java/org/jgrasstools/hortonmachine/modules/hydrogeomorphology/saintgeo>

[com/moovida/jgrasstools/tree/master/hortonmachine/src/main/java/org/jgrasstools/hortonmachine/modules/hydrogeomorphology/saintgeo](https://github.com/moovida/jgrasstools/tree/master/hortonmachine/src/main/java/org/jgrasstools/hortonmachine/modules/hydrogeomorphology/saintgeo)

RIVER PROFILE

supercritical flow



Project: MapServer

Student: Samuel Lapointe

Mentor(s): Daniel Morissette; Jessica Lapointe

Title: Add productivity tools to MapServer's ScribeUI



Description: The goal of the project was to improve the error detection system, add the ability to export and import maps and add a way to generate a set of classes from data.

State of the project before GSoC: When I wanted to debug a map that didn't work, I had to check the generated Mapserver file to find the error. To create classes from a set of data, I had to analyse the data and create the classes using bounds I had manually calculated, and the only way to export a map was to use Git.

Addition to the project: Now, I can see most syntax errors directly in the editor's interface, I can use a menu to generate classes for a layer and I can export or import maps to share them with other users.

Links: Project page: <https://github.com/mapgears/scribeui>

Getting started: <https://github.com/mapgears/scribeui/wiki/Getting-started-with-ScribeUI>

Demo: <http://demo.scribeui.org/>

ScribeUI

Class generator

Improved error detection

ScribeUI v1.3 interface showing a map configuration editor. The editor displays a QML-style configuration for a map layer. A syntax error is highlighted in a yellow box: "loadStyle(): Unknown identifier. Parsing error near (,):(line 651)". The configuration includes settings for data source, field, class type, mode, number of classes, and colors. It also defines a style for the 'water' layer, including color, width, and label settings.

```
42 @layerconfig
43 DATA {
44     5-10:
45     '50m_physical/ne_50m_rivers_lake_centerlines_scale_rank'
46     11-16:
47     '10m_physical/ne_10m_rivers_lake_centerlines_scale_rank'
48     LABELITEM: "name"
49     CLASS {
50         EXPRESSION: ([strokeweight] >= 0 AND [strokeweight] <=
51         0.2)
52         STYLE {
53             COLOR: @rivercolor
54             WIDTH {
55                 5-6: 1,5
56             }
57             7-16: 3
58         }
59         LABEL {
60             6-16 {
61                 COLOR: @watercolor_lbl
62                 OUTLINECOLOR: 255 255 255
63                 FONT: 'scb'
64                 TYPE: truetype
65                 SIZE: 10
66                 REPEATDISTANCE: 100
67                 MINDISTANCE: 300
68                 ANGLE: FOLLOW
69                 POSITION: AUTO
70                 PARTIALS: FALSE
71             }
72         }
73     }
74     CLASS {
75         EXPRESSION: ([strokeweight] >= 0.200000001 AND
76         [strokeweight] <= 0.35)
77         STYLE {
```

The 'Classify' dialog box in ScribeUI. It allows users to configure a classification for a map layer. The 'Group/Layer' is set to 'admin'. The 'Data source' is '10m_cultural/ne_10m_admin_'. The 'Field' is 'pop_est'. The 'Class type' is 'Sequential'. The 'Mode' is 'Quantile (Equal Count)'. The 'Number of classes' is 10. The 'Colors' are set to '#e6e6ff,#ccccf0,#b3'. A table on the right shows the distribution of values for the 'pop_est' field.

Color	Lower bound	Upper bound	Occu
-99	7850	25	
7850	70000	26	
70000	306694	25	
306694	1514993	26	
1514993	4119083	25	
4119083	6995655	25	
6995655	10625176	26	
10625176	22215421	25	
22215421	48508972	26	
48508972	1338612970	24	

Map export and import

The 'Export map' dialog box in ScribeUI. It allows users to export a map. The dialog includes a section for selecting the amount of vector and raster data to include. The options are: 'No data' (selected), 'Minimal data', and 'All data'. The 'Status' is 'Not started'.

Select the amount of vector and raster data to include. If you placed your data outside the pdata folder, these options will not work properly.

☒ No data Do not include any raster or vector data. A user who wishes to view the map will need to have the appropriate data already installed.

☐ Minimal data Include only the data necessary for the map to render, no unused data will be included.

☐ All data Include all data, even vector or raster files not used in the map. This option will probably create very large files.

Logs

Status Not started

Project: Opticks

Student: Tom Van den Eynde

Mentor(s): Trevor Clarke

Title: Image Enhancement/Background Suppression



Description: The goal of the project was to implement the image enhancement algorithm Drizzle for both image and video input.

State of the project before GSoC: In Opticks, when I wanted to use a low resolution image I had to resort to other software to first improve the quality or reduce the noise interference.

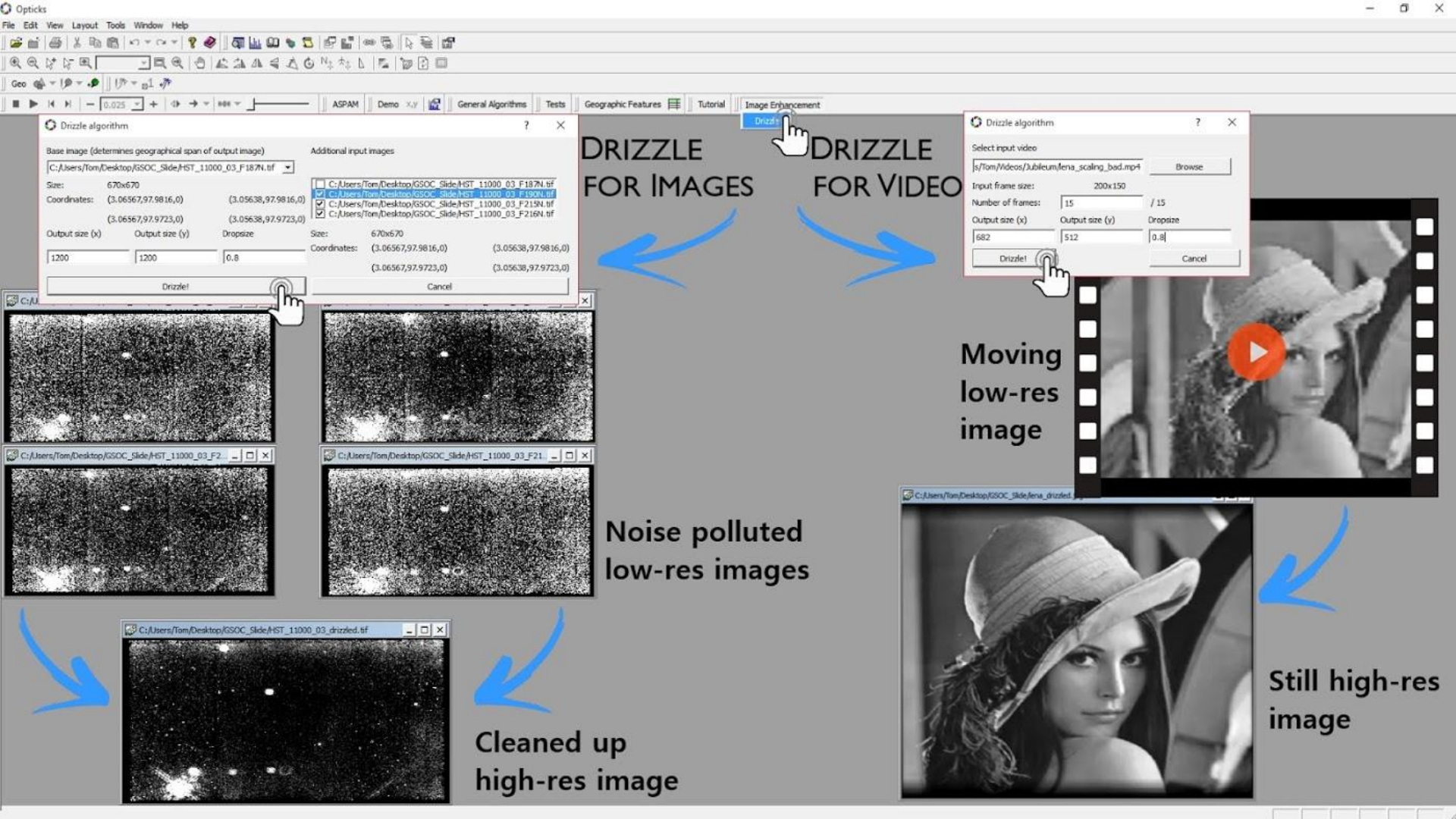
Addition to the project: With the addition of the Drizzle algorithm, Opticks can enhance and improve the interpretability of images. Now, even low resolution or heavily noise polluted images can be used for various remote sensing applications.

Links: Project: <https://www.google-melange.com/gsoc/project/details/google/gsoc2015/tomvde/5733935958982656>

Blog: <http://drizzle-for-opticks-gsoc2015.blogspot.be/>

Repository: <https://github.com/Tom-VdE/Drizzle-plugin>

Instructions: <https://github.com/Tom-VdE/Drizzle-plugin/blob/master/README.md>



Project: Open Trip Planner

Student: Nipuna Gunathilake

Mentor(s): Sean Barbeau, Stefan Steiniger

Title: GTFS-Realtime validation tool for Open Trip Planner

Description: This application can be used to validate feeds of General Transit Feed Specification Real Time (GTFS-RT) feeds.

State of the project before GSoC: GTFS and GTFS-realtime have become dominant formats for open data in the transit industry. While the GTFS format has a GTFS Feed Validator, no such open-source tool currently exists for GTFS-rt.

Addition to the project: This project would implement a tool that takes a GTFS and GTFS-rt feed as input, and examines the GTFS-rt feed to determine if it properly meets GTFS-rt best practices, both those explicitly listed in the spec and those based on the needs of consuming applications.

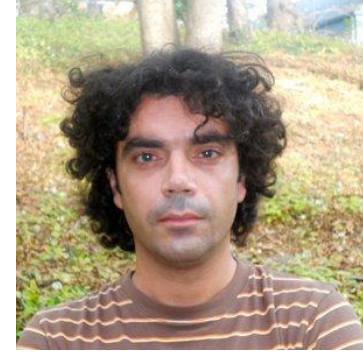
Links: <https://github.com/CUTR-at-USF/gtfs-realtime-validator/wiki>

Project: OSGeo-Live

Student: Massimo Di Stefano

Mentor(s): Cameron Shorter, Rich Signell, Angelos Tzotsos

Title: Integration of geospatial OSS in educational notebooks



Description: This GSoC 2015 Idea will focus on the development of a cross-projects python library with the aim of bridging together the several software libraries already installed on the live through the use of the Jupyter notebook server in a series of "topic-oriented" geospatial notebooks.

State of the project before GSoC: before this GSoC there was no jupyter-notebook installed and very limited set of example based on the IPython-notebook.

Addition to the project: After GSOC 2015 an installer for the jupyter-notebook has been developed (enabling multiple kernels [py2, py3, R, Bash, Julia]) and a series of geospatial notebooks to show the usage of several project like GDAL, OSSIM, GRASS has been added. The project also included the building on Debian packages for the jupyter single user ecosystem and the collaborative development of widget for interactive visualization of geospatial data from a notebook ([CesiumWidget](#))

Links: GSoC Proposal: https://wiki.osgeo.org/wiki/Live_GIS_GSoC_2015

Weekly reports: https://github.com/epifanio/IPython_notebooks/tree/master/OSGeo-live/osgeolive-gsoc-2015/Report

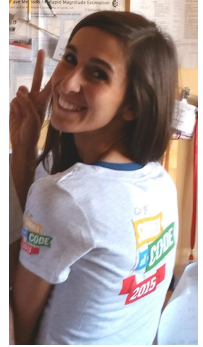
Repository: <https://github.com/epifanio/OSGeoNotebooks> - Video: <http://epinux.com/cesium-720p.mp4>

Project: OSSIM

Student: Martina Di Rita

Mentor(s): Oscar Kramer, David Burken

Title: Tool for DSM generation using tri-stereo optical imagery



Description: the work has been based on a previous GSoC project through which a tool, named DATE, for Digital Surface Model (DSM) generation from optical stereo-pair was developed within OSSIM framework. The idea was to improve and update DATE in order to generate DSMs from tri-stereo optical imagery and fix some bugs in the DSMs generation processing chain. DATE key features include the use of computer vision algorithms in order to improve the processing efficiency and make the DSMs generation process fully automatic.

State of the project before GSoC: before this GSoC it was not possible to process tri-stereo imagery and there was still some problem in finding the right quasi-epipolar geometry in the stereo version in order to compute a proper Disparity Map.

Addition to the project: with DATE plug-in it is now possible to generate DSMs within OSSIM framework starting from stereo and tri-stereo satellite optical imagery. Furthermore, from the stereo processing side, I added an image preprocess filtering and I found a better solution for the quasi-epipolar geometry achievement.

Links: GSoC Proposal: <https://www.google-melange.com/gsoc/proposal/review/student/google/gsoc2015/martidi/5717271485874176>

Weekly reports: <https://trac.osgeo.org/ossim/wiki/OSSIMtoolForDSMgenerationFromTristereoAndSARimagery>

Please test my code in https://github.com/martidi/opencv_dsm/tree/master following the instructions in README.md file

Project: pgRouting

Student: Sarthak Agarwal

Mentor(s): Daniel Kastl, Vicky Vergara

Title: New osm2pgrouting import tool to import OpenStreetMap (OSM) data in pgRouting



Description: The idea for this year GSOC project was to update the existing osm2pgRouting tool which helps users to import open source osm data into pgRouting database tables.

State of the project before GSoC: Before the gsoc project the tool lack support for boost library and the code was to be refactored and commented. The code was not documented and the tool on a whole needed documentation(WIKI pages)

Addition to the project: After GSOC 2015 support for boost library such as boost.program_options and boost.geometry has been added to the software. Now user can give options in any order while running the software and boost will do the parsing. The code was not commented and refactored, which was updated. Several wiki pages are added to better understand the tags used for routing present in osm data and all the attributes related to those tags. The link to the wiki page can be found below.

Links: Weekly reports : <https://github.com/pgRouting/pgrouting/wiki/GSoC-2015-Sarthak-Agarwal>

Repository : <https://github.com/sarthak-0415/pgrouting/tree/sarthak-0415>

Project: PyWPS-4

Student: Calin Cristian Nicolae Ciociu

Mentor(s): Jachym Cepicky, Luís de Sousa

Title: REST Interface



Description: My project added the ability to use a REST interface to control, add, remove processes and change configuration of PyWPS-4 instances.

State of the project before GSoC: We had to create/delete the processes by accessing the storage of the server which required a restart of the server every time something got changed.

Addition to the project: With the addition of the REST interface I wrote everyone who needs to update, delete, or add new processes and change configuration can do so without accessing the server.

Links: Project: <http://www.google-melange.com/gsoc/project/details/google/gsoc2015/noctalin/5717271485874176>

Repository: <https://github.com/jachym/pywps-4>

Instructions: <https://github.com/PyWPS/pywps-4-demo/blob/master/README.rst>

PyWPS-4 REST interface

This is a HOWTO for the REST interface of PyWPS.

	GET	PUT	POST	DELETE
http://localhost:5000/wpsadmin/configuration	Returns a list of all server instance configuration	-	-	-
<a href="http://localhost:5000/wpsadmin/configuration/<int:server_id>">http://localhost:5000/wpsadmin/configuration/<int:server_id>	Returns the configuration of serverid server instance	Changes specific configuration of serverid server instance. (Passed data must be JSON formatted and submitted with Header Content-Type application/json)	-	-
http://localhost:5000/wpsadmin/server	Returns a list of server instances containing information about the host, port, pid and process status	-	-	-
<a href="http://localhost:5000/wpsadmin/server/<int:server_id>">http://localhost:5000/wpsadmin/server/<int:server_id>	Returns the host, port, pid and process status of serverid server instance.	Creates and starts a new serverid server instances specified by the passed data. (Passed data must be JSON formatted and submitted with Header Content-Type application/json)	-	Stops and removes serverid server instances from currently available server instances.
<a href="http://localhost:5000/wpsadmin/server/<int:process_id>/process">http://localhost:5000/wpsadmin/server/<int:process_id>/process	Get list of all the processes activated on	-	-	Deactivate/Remove all the processes from the specified

*Displays HOWTO site
for the REST interface*

localhost:5000/wpsadmin/configuration/30910

```
{
  "abstract": "See https://github.com/jachym/pywps-4 and http://www.cenangeospatial.org/standards/wps",
  "administrativearea": "False",
  "city": "Belval",
  "constraint": "NONE",
  "constraints": "NONE",
  "contactinstructions": "NONE",
  "country": "LU",
  "debug": "true # deprecated since 3.2, use logLevel instead",
  "deliverypoint": "Street",
  "electronicaddress": "login@server.org",
  "electronicmailaddress": "login@server.org",
  "encoding": "utf-8",
  "fees": "NONE",
  "hoursofservice": "00:00-24:00",
  "individualname": "Calin Ciociu",
  "keywords": "GRASS, GIS, WPS",
  "lang": "en-CA",
  "logfile": "",
  "loglevel": "INFO",
  "maxfilesize": "3mb",
  "maxinputparamlength": "1024",
  "maxoperations": "30",
  "maxrequestsize": "3mb",
  "maxsingleinputsize": "1mb",
  "onlineresource": "http://foo.bar",
  "outputpath": "/tmp/outputs/",
  "outputurl": "/data/",
  "phonefacsimile": "False",
  "phonevoice": "False",
  "positionname": "Intern",
  "postalcode": "000 00",
  "processespath": "",
  "providername": "LIST",
  "providersite": "http://foo.bar",
  "role": "Developer",
  "serveraddress": "http://localhost",
  "serverport": "5000",
  "temppath": "/tmp/pywps/",
  "title": "PyWPS 4 Server",
  "version": "1.0.0",
  "workdir": "/tmp"
}
```

*Retrieve the configuration
of the PyWPS instance with
PID 30910*

*Retrieval, addition, update, deletion
are done by using GET, PUT, POST, DELETE
HTTP methods*

localhost:5000/wpsadmin/server/process

```
{
  "processes": {
    "0": "sleep.py",
    "1": "area.py",
    "2": "centroids.py",
    "3": "ultimate_question.py",
    "4": "feature_count.py",
    "5": "buffer.py",
    "6": "sayhello.py"
  }
}
```

*Get the list of all available
PyWPS processes*

Project: QGIS

Student: Marcus Santos

Mentor(s): Victor Olaya, Alexander Bruy

Title: Multithread Support on Processing Toolbox



Description: This project consisted on making the QGIS Processing toolbox multi-threaded, allowing to have non blocking analysis/geoprocessing jobs, prevent crashes (when a QGIS module crashes for some reason) and also improve the user experience.

State of the project before GSoC: Before the multithreading support, the user was only able to run one analysis at a time and there was no option to cancel the algorithm execution.

Addition to the project: With multithreading support the user is allowed to cancel long geoprocessing jobs and also run several algorithms at the same time.

Links: Project page: <https://qgisgsoc2015.wordpress.com>

Repository: <https://github.com/mvcsantos/QGIS>

Demo: <https://www.youtube.com/watch?v=V-CJCnKw89U>

Main thread

Thread Y

Thread X

New option to
cancel the algorithm
execution

Cancel

Fechar

Run

For more info, please visit :

https://wiki.osgeo.org/wiki/Google_Summer_of_Code_2015_Results