Open Source GIS Blitz!


Start with this video
http://vimeo.com/2598878

Openstreetmap.org

Crown copyright– no expiration – all government map data in England.

Rest of Europe not much better

Volunteer mapping effort.

Bulk uploads of GPS data, vector data sets, online editing using potlatch tool.

Open Geospatial Consortium (OGC) Data Standards

WMS – Web Map Service

WFS-T - Web Feature Service.

WCS – Web Context Service

SOS – Sensor Observation Service

KML - Keyhole Markup Language

WPS – Web Processing System

Other Spatial Standards

GeoRSS

GeoJson


Libraries

Other Useful Libraies

Geos - http://geos.refractions.net

Java Topology Suite
http://www.vividsolutions.com/jts/jtshome.htm

Liblas - http://www.liblas.org - library for working with LAS format Lidar data

Gdal - http://www.gdal.org

“Swiss army knife” of geospatial data

Features:

Interfaces for 61 Raster formats and 27 vector formats.

Reprojection of raster and vector datasets.

Merging and splitting raster and vector datasets.

Basic library used in many Open source GIS projects.

Cross-platform easy install Fwtools binaries with command line utilites

Library interfaces to the following programming/scripting languages:

Visual Basic 6 (no swig)

C/C++

Perl

Python

Java – testing

Ruby – testing

C# - testing

R statistical programming

You may already be using gdal in your current GIS software! The license allows for use in open source and commercial products. Used in :

GRASS

UMN Mapserver

QGIS

ESRI ArcGIS 9.2+

Google Earth

and many more.

Scripting Image Manipulation

A case history – Converting the 1 image per County NAIP UTM MrSID imagery to a Seamless Layer of DOQQ Imagery in North Carolina State Plane Projection.

Doug Newcomb
USFWS
December, 2007

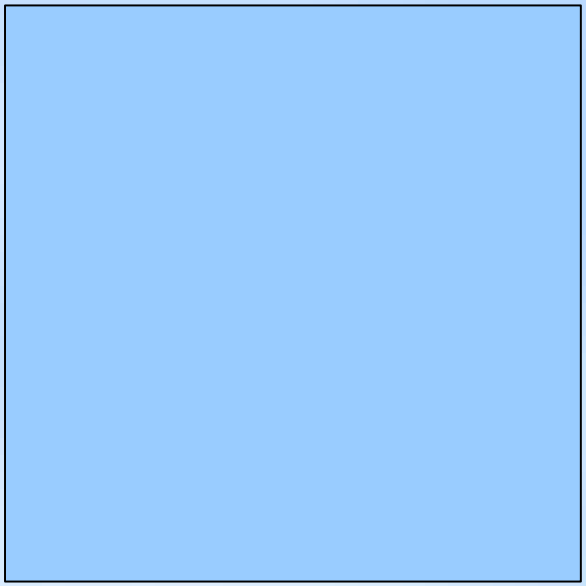
Hardware and software

Hardware: USFWS Standard Dell Power User Intel Core 2 Duo Workstation with 6GB RAM, 1x 80 GB SATA Hard drive, 1x 750 GB SATA hard drive, 1x 500GB USB 2.0 hard drive

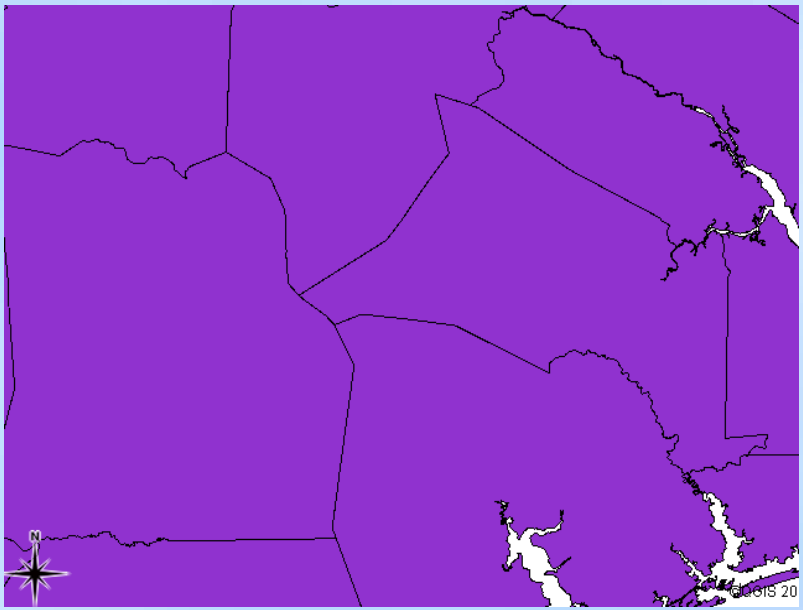
Software: Centos 5.0 64-bit Linux, Gdal (compiled with ECW library), FWTools (http://fwtools.maptools.org)

Problem Description 1

Raster Imagery is Rectangular

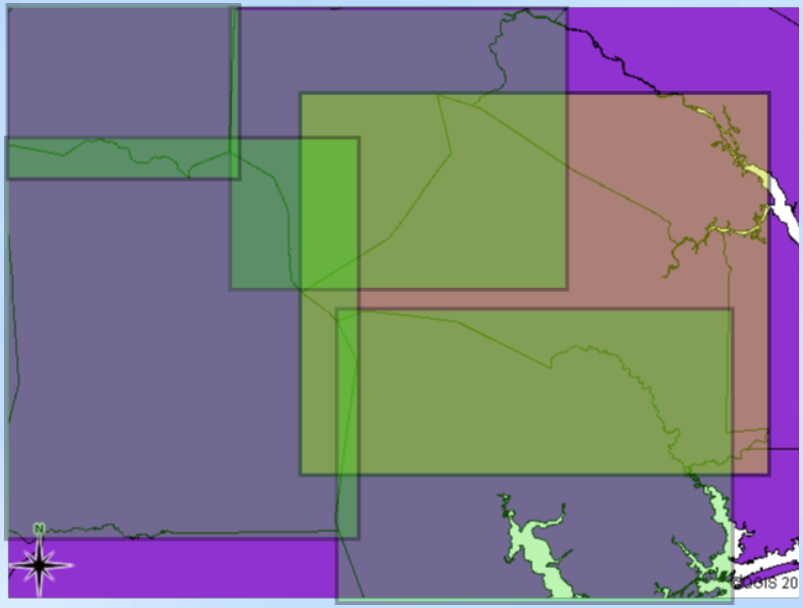


Counties are usually not Rectangular



Problem Description 1a

Working with more than one County can



Problem Description 2

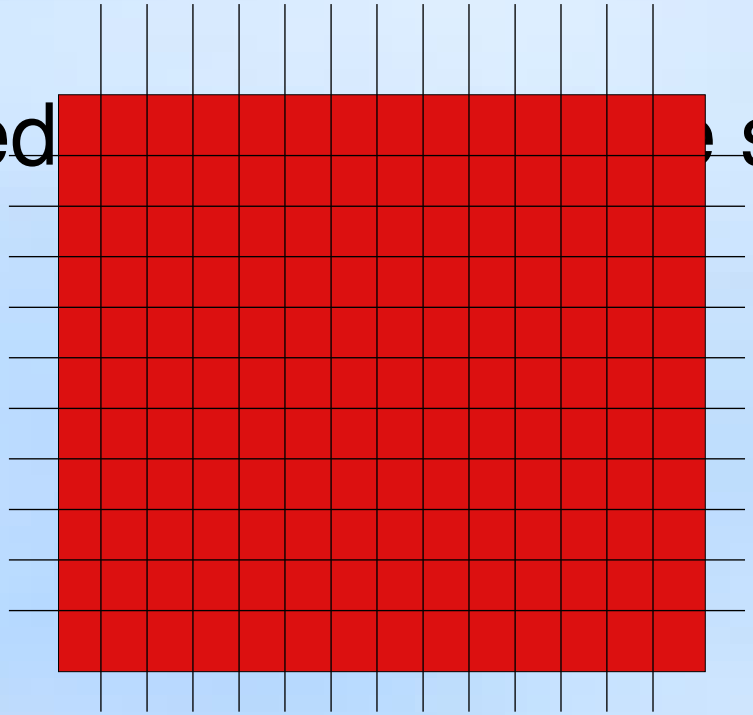
ones so much, we have 3 of them! Unfortunately UTM zo

e NC State Plane projection is standard for all of our stat

gle projection in a regular tiled mosaic. The ideal would b

Reprojection of Imagery

y pixels are aligned e system of the proje



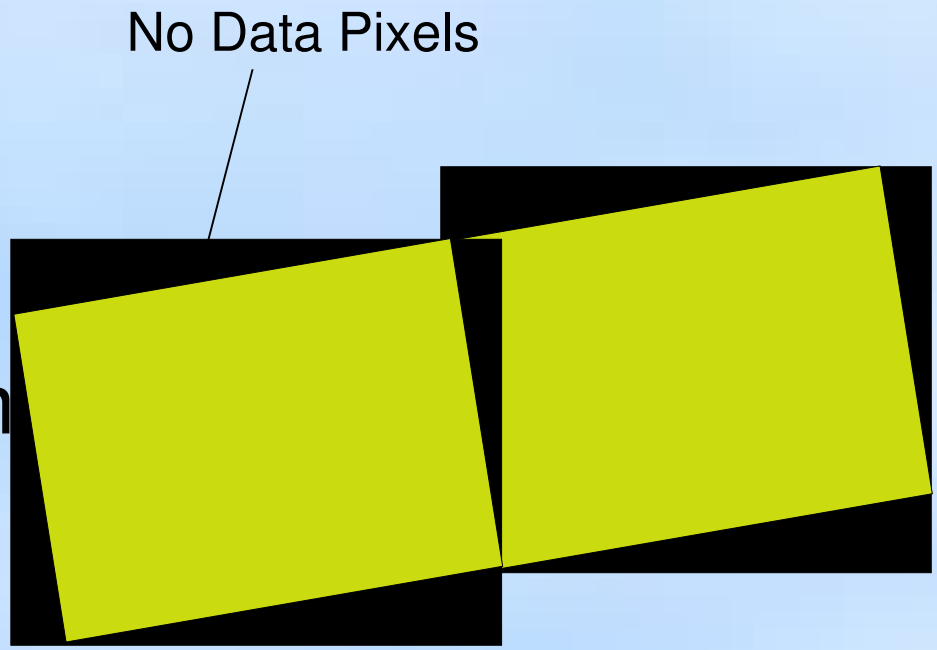
Reprojection of Imagery

ntally aligned to the coordinate system of the new project

No Data Pixels

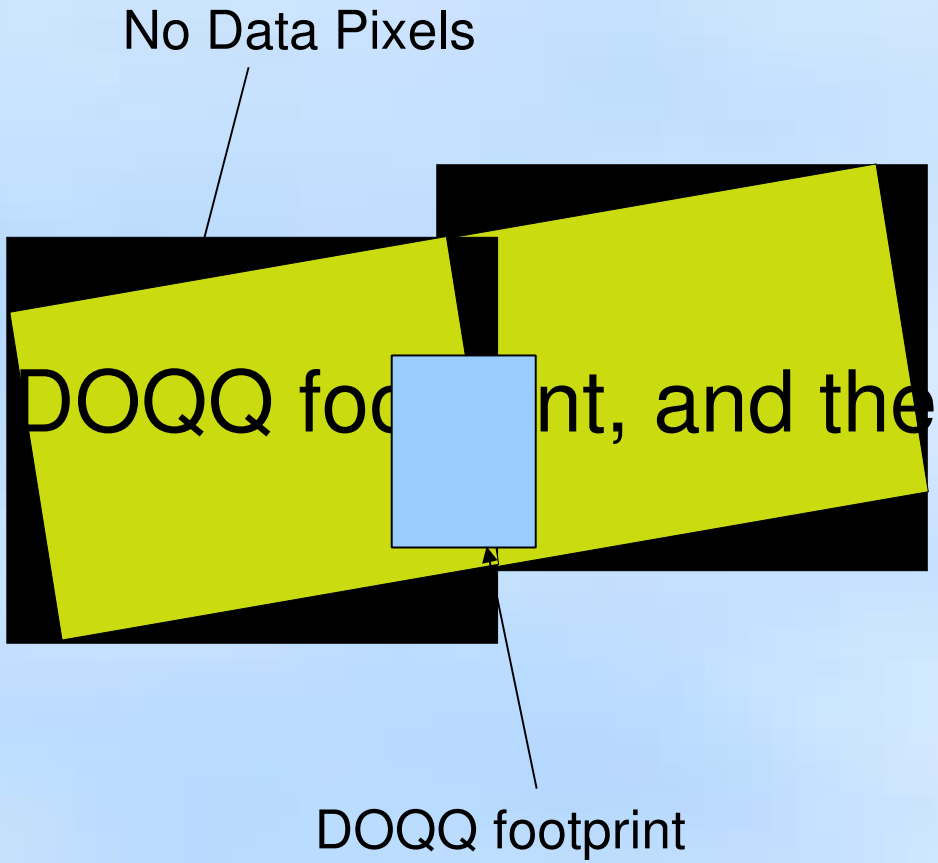
Reprojection of Imagery

lt when multiple reprojected im



Cookie cutting

art of each file that will match a DOQQ fo nt, and the



Solution: A free, open source tool and a bit of python scripting

hat allow for the reprojection, merging and reformatting of

Python scripting in GDAL

w you to treat images as python objects that can be man

do that. Instead I used python to repetitively substitute va

ogr2ogr
gdal_translate

First Step: Converting the Mr.Sid imagery to something useful

nvert the MrSID files. The bulk of the processing was done with the 6

nd set the output format to be Erdas Imagine. It took about 36 hours

Linux filesystem tricks

al with all of the data as a single unit and have enough working space

x soft link

s feature has been available in Unix style systems for about 20 years

Getting DOQQ corner coordinates

olina, I ran the gdaltindex command on the existing 1998 DOQQ laye

s the Wildcard pattern. The filename of each image is recoded in the

a shapefile to a flat text file.

r the shape file, which I redirect to a text file.

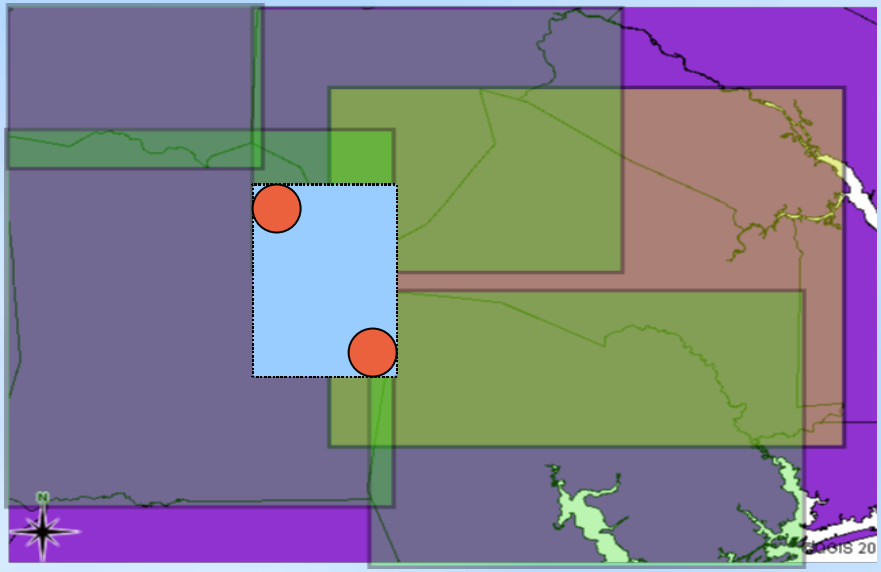
Getting DOQQ corner coordinates

n index of the Erdas Imagine files (which is why we needed the imag

LL,UR,LR coordinates and the filename is gathered using standard py

Finding the Intersection

of the first polygon are used to select the footprints of the Imagine file



Clipping the Imagine files

and is used to extract the portion of the image that is within the area o

enerated geotiff files into one geotiff file.

Compressing the Final File

en converted to an ECW file and the tiff world file is copied to the mor

```
slate -a_srs \"epsg:32119\" -of \"ECW\" -c
 % (doqtifname,doqecwname)

doqtfw,doqwld)
```

ly 3500 times.

Geotools - http://geotools.codehaus.org/

Open Source Java Code Library that provides standards-compliant manipulation of geospatial data. Primarily used in Udig, and Geoserver.

Geotools - http://geotools.codehaus.org/

Data Sources: Shapefile, GML, WFS, Oracle Spatial, ARCSDE, MySQL, Geomedia, TIGER, PostGIS, VPF, Mapinfo, ArcGRID, AsciiGRID, Raw image (with World File), GeoTIFF* , WMS*

* - In Development

Servers and Services

Postgresql Features and Info

(From the website)

Open source relational database system.
15 years of active development
Strong reputation for reliability, data integrity, and correctness.
Runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows.
Fully ACID compliant, has
Full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It
Includes most SQL92 and SQL99 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, and TIMESTAMP.
Supports storage of binary large objects, including pictures, sounds, or video.
Native programming interfaces for C/C++, Java, Perl, Python, Ruby, Tcl, ODBC, among others, and exceptional documentation.
Multi-Version Concurrency Control (MVCC),
point in time recovery,
tablespaces

Postgresql Features and Info

Asynchronous replication,
Nested transactions (savepoints),
Online/hot backups
Sophisticated query planner/optimizer
Write ahead logging for fault tolerance.
International character sets,
Multibyte character encodings, Unicode,
Locale-aware for sorting, case-sensitivity, and formatting. It is
Highly scalable both in the sheer quantity of data it can manage and and in the number of concurrent users it can accommodate.

Postgresql

Current Limitations:
Maximum Database Size Unlimited*
Maximum Table Size 32 TB
Maximum Row Size 1.6 TB
Maximum Field Size 1 GB
Maximum Rows per Table Unlimited
Maximum Columns per Table 250 - 1600 depending on column types
Maximum Indexes per Table Unlimited

Postgresql

Programming Interfaces:

- **Perl**
- **Python**
- **Php**
- **Ruby**
- **Java**
- **C**
- **C++**
- **.Net**
- **ADA**
- **Smalltalk**
- **ODBC**
- **TCL**
- **Bash**

Postgis

Postgis is a spatial data extension to the Postgresql database server.

Postgis creates spatial query/data types in Postgresql which allow any language/ that has an interface to the database to perform SQL queries to perform spatial queries.

Postgis follows the OGC Simple Features Specification for SQL for all geometry types and methods found at http://www.opengeospatial.org/docs/99-049.pdf

PostGIS extends the OGC standard with support for 3DZ,3DM and 4D coordinates.

Postgis – Sample Spatial SQL

The actual SQL query in php takes the form

This section makes the initial database connection

```
<?$db=pg_connect("host=192.168.0.46 port=5432 user=******* password='*******' db
name=ncdata");
?>
```

---

spatial query section

```
$dbquery="select distinct r.com_name,r.fed_stat from nheo_pt as r, cb_dot03 as m
where (r.fed_stat!='') and (m.co_name='"'". $_POST['county'] ."'" .") and
((r.the_geom && m.the_geom ) AND (distance(m.the_geom,r.the_geom)< 1))";
```

This section makes it pretty on a web page

```
$result = pg_query($db,$dbquery);
   $rows = pg_numrows($result);
   for ($i=0; $i<$rows; $i++)
   {
      $data = pg_fetch_object($result, $i);
?>
```

Postgis – Sample Spatial SQL

To break the above query down:
select distinct r.com_name,r.fed_stat - get the unique rows from the returned common name and federal status .

from nheo_pt as r, cb_dot03 as m - the nheo_pt table is the Natural Heritage Program Element Occurance dataset (loaded directly from the shape file into the database) and cb_dot03 table is the North Carolina County Boundary dataset (loaded directly from the shape file into the database) - we make an alias of r for the nheo_pt and m for the cb_dot03 table
where - start of conditions of comparison

(r.fed_stat !='') and(m.co_name='".'"'. $_POST['county'] .'"' .") - fed_stat is the federal status field in the imported Element Occurance shape file. !='' means to get every entry where the federal status is not blank. co_name is the County Name field in the imported County shape file . The $_POST['county'] entry passes the county name selected from the drop-down menu in the form to the SQL query.

and ((r.the_geom && m.the_geom) - this is a quick comparison of the rectangular bounding box of the 2 geometries that have been selected, basically making a quick comparison of the extents of the geometries selected in the initial select statement.

AND (distance(m.the_geom,r.the_geom)< 1))"; - this section looks at the detailed geometry of each table to make a precise selection based on the linework of the boundary of the selected County polygon and the Element Occurance points to return the final selected rows in the

Postgis – Sample Spatial SQL

(Example From the Postgis website - http://postgis.refractions.net)

What is the length of roads fully contained within each municipality?

This is an example of a "spatial join", because we are bringing together data from two tables (doing a join) but using a spatial interaction condition ("contained") as the join condition rather than the usual relational approach of joining on a common key:

```
postgis=# SELECT m.name, sum(length(r.the_geom))/1000 as roads_km
        FROM bc_roads AS r,bc_municipality AS m
        WHERE r.the_geom && m.the_geom
        AND contains(m.the_geom,r.the_geom)
        GROUP BY m.name
        ORDER BY roads_km;

          name          |     roads_km
------------------------+------------------
 SURREY                 | 1539.47553551242
 VANCOUVER              | 1450.33093486576
 LANGLEY DISTRICT       | 833.793392535662
 BURNABY                | 773.769091404338
```

Postgis – Sample Spatial SQL

(Example From the Postgis website - http://postgis.refractions.net)

Create a new table with all the roads within the city of Prince George.

This is an example of an "overlay", which takes in two tables and outputs a new table that consists of spatially clipped or cut resultants. Unlike the "spatial join" demonstrated above, this query actually creates new geometries. An overlay is like a turbo-charged spatial join, and is useful for more exact analysis work:

```
postgis=# CREATE TABLE pg_roads as
        SELECT intersection(r.the_geom, m.the_geom) AS intersection_geom,
              length(r.the_geom) AS rd_orig_length,
              r.*
        FROM bc_roads AS r, bc_municipality AS m
        WHERE r.the_geom && m.the_geom
        AND intersects(r.the_geom, m.the_geom)
        AND m.name = 'PRINCE GEORGE';
```

Postgis – Sample Spatial SQL

(Example From the Postgis website - http://postgis.refractions.net)

What is the length in kilometers of "Douglas St" in Victoria?

```
postgis=# SELECT sum(length(r.the_geom))/1000 AS kilometers
        FROM bc_roads r, bc_municipality m
        WHERE r.the_geom && m.the_geom
        AND r.name = 'Douglas St'
        AND m.name = 'VICTORIA';
   kilometers
-----------------
 4.89151904172838
(1 row)
```

Postgis – Sample Spatial SQL

Postgis will return the geometry of the vector data as :

WKT (Well Known Text),

WKB (Well Known Binary)

SVG (scalable vector graphics) – This brings up the intriguing possibility of the database returning vector data directly to the browser WITHOUT going through a mapserver.

GeoJson – Lightweight vector geometry data

University of Minnesota Mapserver

MapServer is an Open Source development environment for building spatially-enabled internet applications. MapServer is not a full-featured GIS system, nor does it aspire to be. Instead, MapServer excels at rendering spatial data (maps, images, and vector data) for the web. -- from Mapserver website http://mapserver.gis.umn.edu

Mapserver Features:

(Yes, these are copied from the website)

Free and Open Source

Cross platform support

*** Linux, Windows, Mac OS X, Solaris, and more**

Advanced cartographic output

*** Scale dependent feature drawing and application execution**

*** Feature labeling including label collision mediation**

*** Fully customizable, template driven output**

*** TrueType fonts**

*** Map element automation (scale bar, reference map, and legend)**

*** Thematic mapping using logical- or regular expression-based classes**

Support for popular scripting and development environments like PHP, Python, Perl, Ruby, Java, and C#

Mapserver Features:

(Yes, these are still copied from the website)

A multitude of raster and vector data formats

*** TIFF/GeoTIFF, EPPL7, and many others via GDAL**
*** ESRI shapfiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL and many others via OGR**
*** Open Geospatial Consortium (OGC) web specifications**
o WMS (client/server), non-transactional WFS (client/server), WMC, WCS, Filter Encoding, SLD, GML

Map projection support

*** On-the-fly map projection with 1000s of projections through the Proj.4 library**

Mapserver Usage:

Binary cgi-bin executable – uses no memory until a query is made, or can be combined with FastCGI to allow for persistant connections to data sources on high traffic sites. All arguments to the cgi-bin program controlled by .map template file

Use mapscript interface to languages such as perl, python, java, php to add mapserver operations/classes to the language. Allows a map template object to be created on the fly based on user input.

Geoserver - http://docs.codehaus.org/display/GEOS/Home

Geoserver is a Java-based mapserver/web built on the Geotools Java spatial toolkit.

Geoserver allows you to publish the map data as:

- Images – via the WMS interface
- Vector data – via the WFS interface
- Allow users to insert, delete, and update data via the WFS-T interface.

Geoserver - http://docs.codehaus.org/display/GEOS/Home

Features:

OGC Certified WMS 1.1.1 and WFS 1.0 web services.

WFS-T (Transactional) with atomic updates to backend data source.

Vector Data from Oracle, DB2, ARCSDE, PostGIS, Shapefile,Mapinfo*, MySQL*, WFS** - Still in Development

Full feature list at : http://docs.codehaus.org/display/GEOS/Features

Mapguide Open Source - http://mapguide.osgeo.org/

Cross platform open source web mapping solution (but happiest on Windows for now). Formerly Autodesk's Mapguide. Open sourced in 2005 under the LGPL license.

Mapguide Open Source - http://mapguide.osgeo.org/

Features:

Ajax viewer with tiled map display for smooth scrolling (Active X may be required).

Data sources: ESRI SHP, SDF, ARCSDE, MySQL, ODBC. Raster via GDAL libraries. DWF files via free download. WMS and WFS

Hierarchical XML storage of data resources.

Security model with inheritance features

PHP, .Net , Java development environment

Deegree - http://www.deegree.org

Degree is a Java framework for for spatial data infrastructures built to conform to OGC and ISO/TC 211 Standards.

Deegree - http://www.deegree.org

OGC Standards support:

- WMS 1.1.1
- WFS 1.0
- WCS 1.0 (Web Coverage Service)
- CSW (Catalogue Service Web-Profile)

WFS and CSW implimentations have transactional support.

Deegree - http://www.deegree.org

OGC pre Standards support:
- SOS (Sensor Observation Service)
- WTS/WPVS (Web Terrain Service/Web Perspective and View Service)
- WPS (Web Processing Service)
- WFS-G (Geocoding)

Data Sources: PostGIS, Oracle Spatial 9i/10g, Shapefile, ARCSDE, SQL database, WMS, WFS

Clients: Browser based (iGeoPortal) and Desktop (deeJUMP)

PyWPS - http://pywps.wald.intevation.org/

PyWPS stands for Python Web Processing Service, an implimentation of the emerging OGC WPS (Web Processing Service) standard.

PyWPS - http://pywps.wald.intevation.org/

Features:

GRASS, Gdal, Proj, and R used in backend,

Can access any data format open to GDAL and GRASS.

Access via python modules.


Clients

QGIS - http://www.qgis.org

Multi-platform User Friendly GUI with a GRASS backend:

Available for Windows, Mac OSX, and Linux.

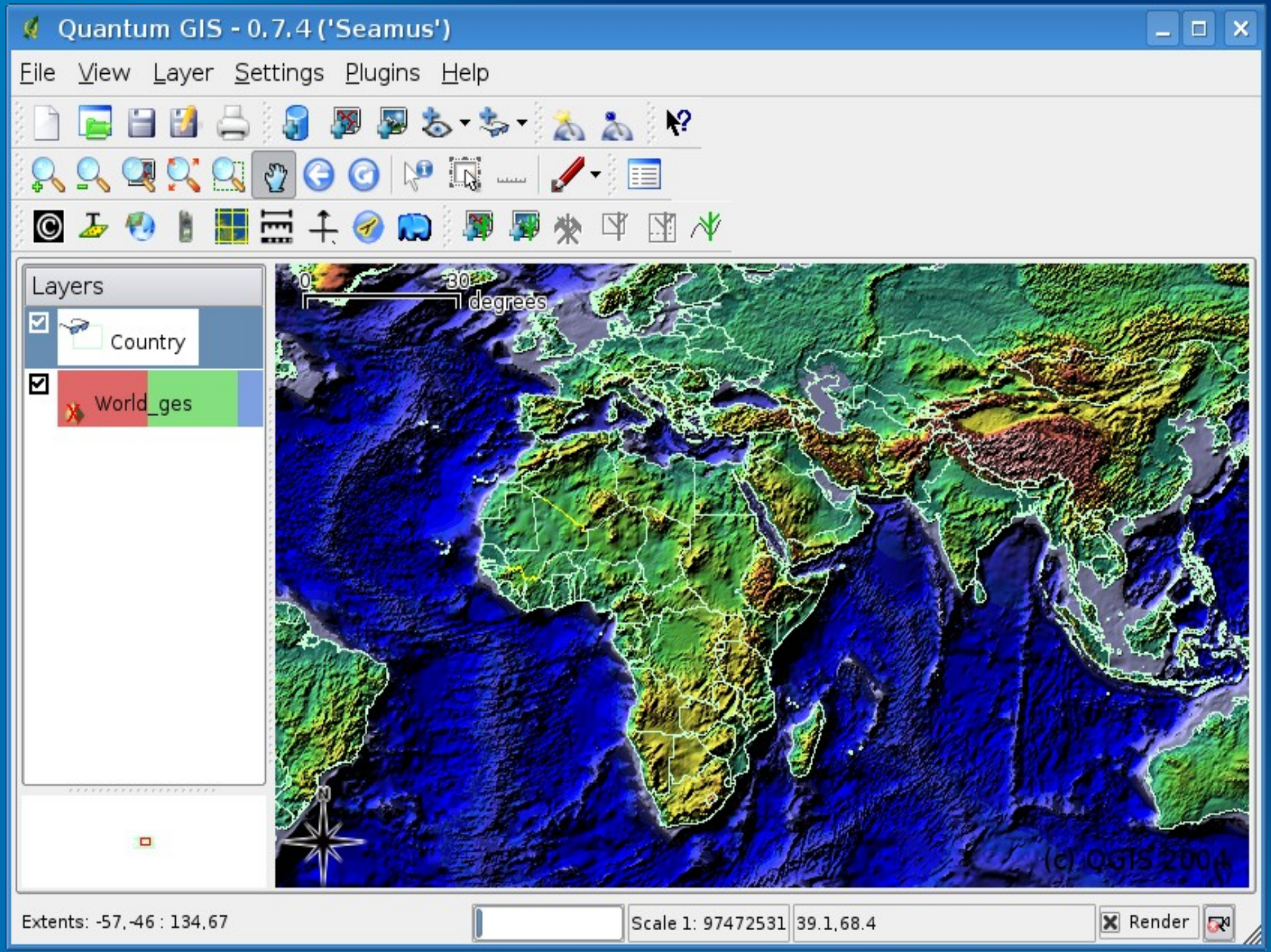
WMS Client

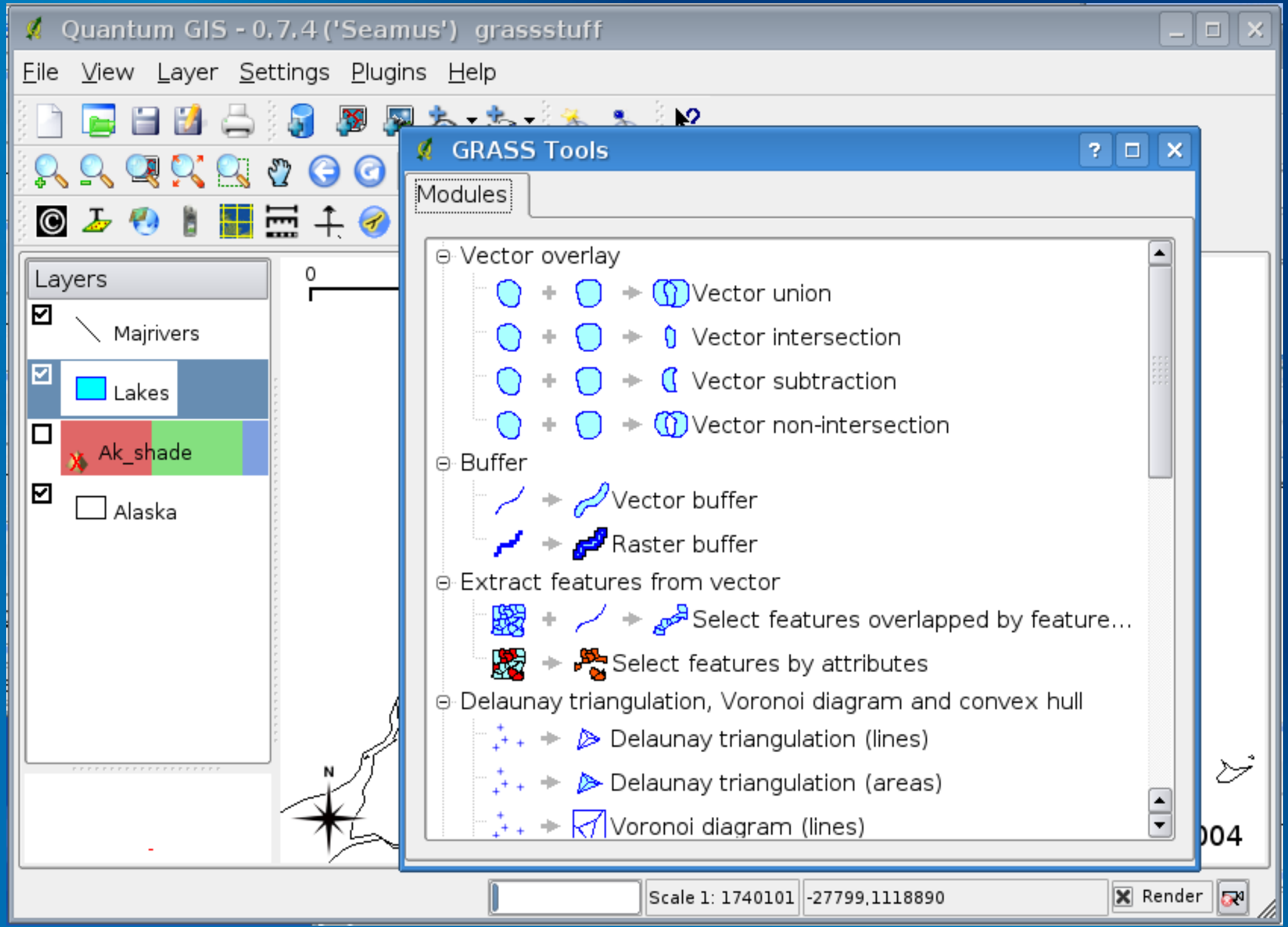
Export view as Mapserver .map template file

Import tool and edit tool for Postgis spatial database.

Raster support for many formats (via GDAL and WMS)

Vector support includes shapefiles, Arc/Info Coverages, Mapinfo files (OGR Vectors).


Quantum GIS - 0.7.4 ('Seamus')
File View Layer Settings Plugins Help
Layers
Country
World_ges
0
30
degrees
(c) QGIS 2008
Extents: -57,-46 : 134,67
Scale 1: 97472531
39.1,68.4
Render


Quantum GIS - 0.7.4 ('Seamus') grassstuff
File View Layer Settings Plugins Help
GRASS Tools
Modules
Vector overlay
Vector union
Vector intersection
Vector subtraction
Vector non-intersection
Buffer
Vector buffer
Raster buffer
Extract features from vector
Select features overlapped by feature...
Select features by attributes
Delaunay triangulation, Voronoi diagram and convex hull
Delaunay triangulation (lines)
Delaunay triangulation (areas)
Voronoi diagram (lines)
Layers
Majrivers
Lakes
Ak_shade
Alaska
0
N
Scale 1: 1740101
-27799,1118890
Render

[Non-Commercial] - Quantum GIS - 0.7.4 ('Seamus')

File View Layer

Layers

Lakes

Alask

[Non-Commercial] - SPIT - Shapefile to PostGIS Import Tool

Shapefile to PostGIS Import Tool

PostgreSQL Connections

test_smp

Connect | New | Edit | Remove

Shapefile List

Add | Remove | Remove All

☐ Use Default SRID

SRID 4326

☑ Use Default Geometry Column Name

Geometry Column Name the_geom

Global Schema public

File Name	Feature Class	Features	DB Relation Name	Schema
C:/WorkSpace/country.shp	MULTIPOLYGON	251	country	public

Help | Import | Close

Scale 1: 857317 421143,1363675 ☑ Render

Udig - http://udig.refractions.net

Udig is a cross-platform, Java – Based GIS desktop application.

Udig - http://udig.refractions.net

Features:

Data Sources: WMS, WFS, WFS-T, Shapefile, Postgis

Drag-n-Drop layer addition

GRASS

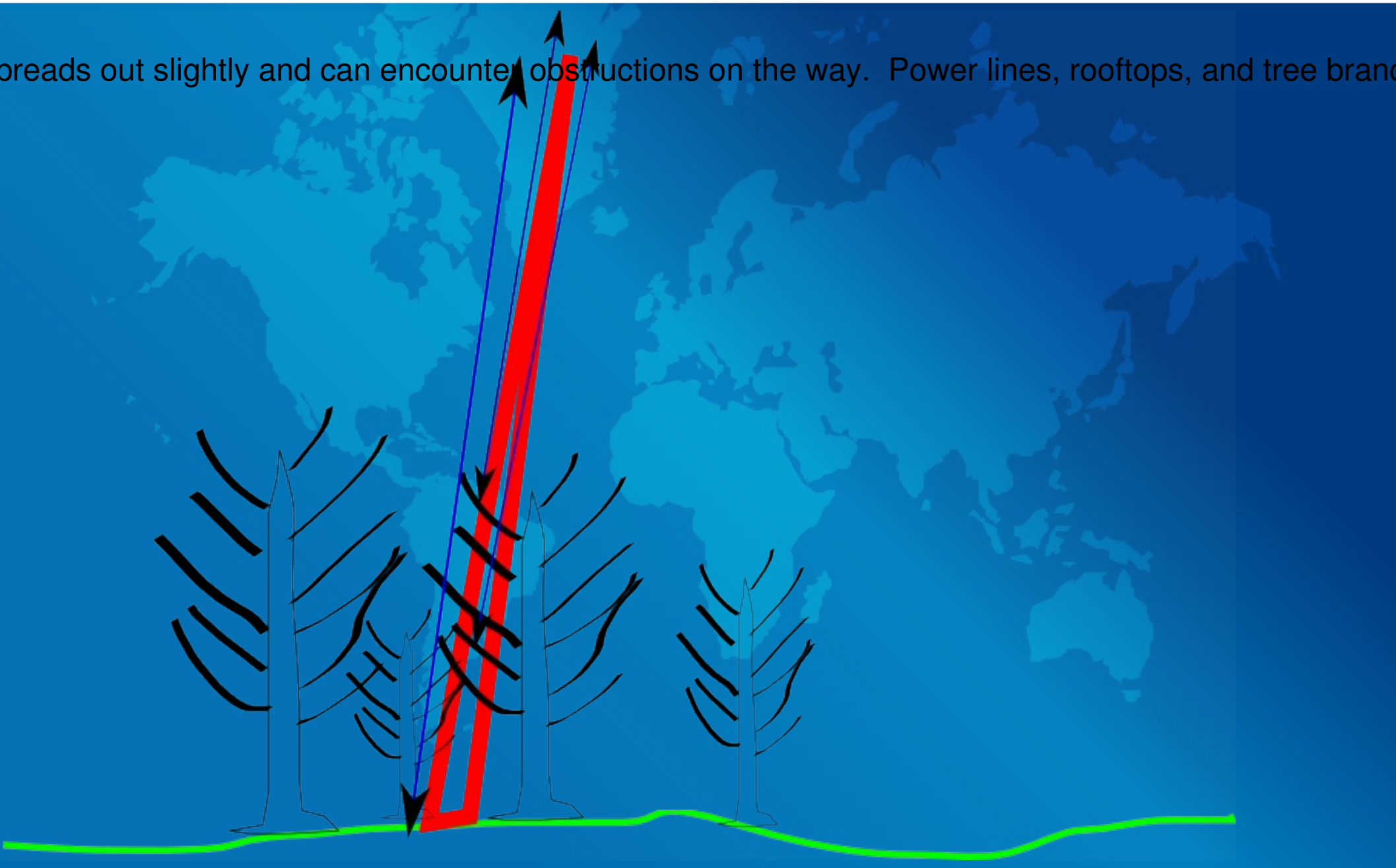
Geographic Resources Analysis Support System (GRASS) was originally developed by the U.S Army Coordinated Engineering Research Lab (CERL) from 1982 -1995 after it was abandoned by the U.S. Government, project development was taken over by academic users worldwide in 1997 and the software license was changed from public domain to GNU GPL . The last CERL release was 4.1 , but the core floating point components for 5.0 were in developed at CERL

GRASS

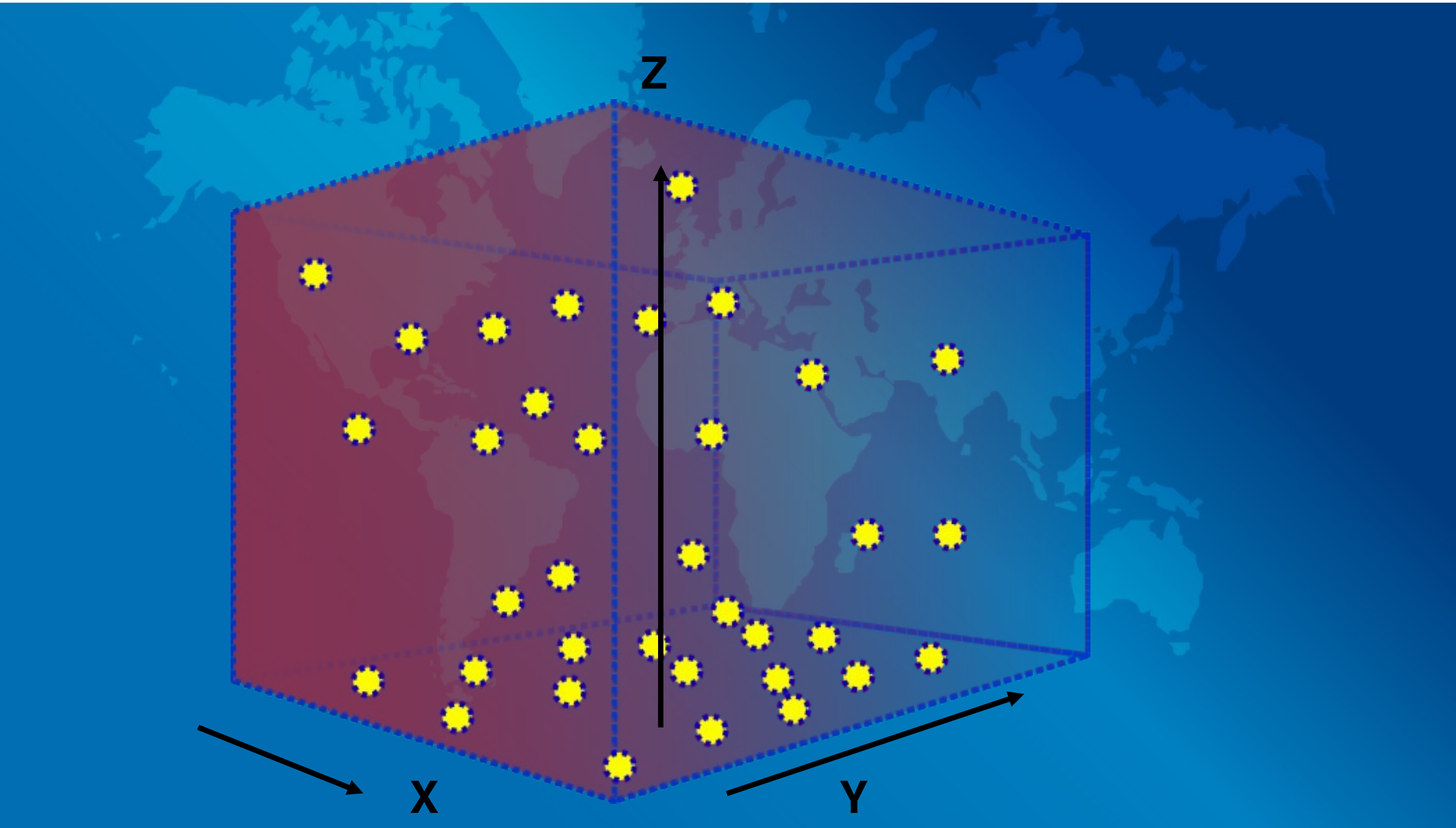
The current release of GRASS is 6.3. Since the last CERL release, there have been several 5.X and 6.X releases. GRASS now has 3 -D vector topology, 3D raster support, 64 bit support for dealing with large datasets, and is in the process of becoming fully cross platform (Windows, OSX, Linux, Solaris) with a wxPython – based GUI .

Lidar Beam Spread and Multiple Returns

preads out slightly and can encounter obstructions on the way. Power lines, rooftops, and tree branc



Lidar Point Cloud



Scaling Problems

Contractor A dataset as aggregated x,y,z is 92 GB file.

Commonly used GIS software on 32-bit Windows is somewhat challenged as input file size demands more than 3.5 GB RAM. Requires 3 stage process to generate surface grids.

GRASS freshly ported to 32-bit Windows, performance vs 64-bit Linux ?

Test!

Test Platform

Hardware:

Dell 755 with Intel Core 2 Quad Q 6700 2.66 GHz processor with 8 GB ram .

OS:

Windows XP SP2 32-bit, Fedora 8 64-bit

Software:

GRASS 6.3 32-bit for Windows

GRASS 6.22 64-bit for Linux (loaded via yum from repo)

Test file size:

32 GB (less than 1/10th of the Phase I Contractor B dataset)

Command:

r.in.xyz input=*fullwacc.txt* output=*wacc10m_num* method=*n* type=F*CELL* fs=, x=*1* y=*2* z=*3* zrange=*0,500* percent=*x*

X=5,20 on Windows, 50 in Linux , 10 m cell size

Test results

32-bit Windows GRASS at 5% of Map in memory:

48 hours 30 minutes

32bit Windows GRASS at 20% of Map in Memory:

8 hours 30 minutes

32-bit Windows GRASS would not run at 50% of map in memory.

64-bit Linux 64 bit GRASS at 50% of map in memory:

Woot!

64 bit Linux / 64bit GRASS > 5x faster

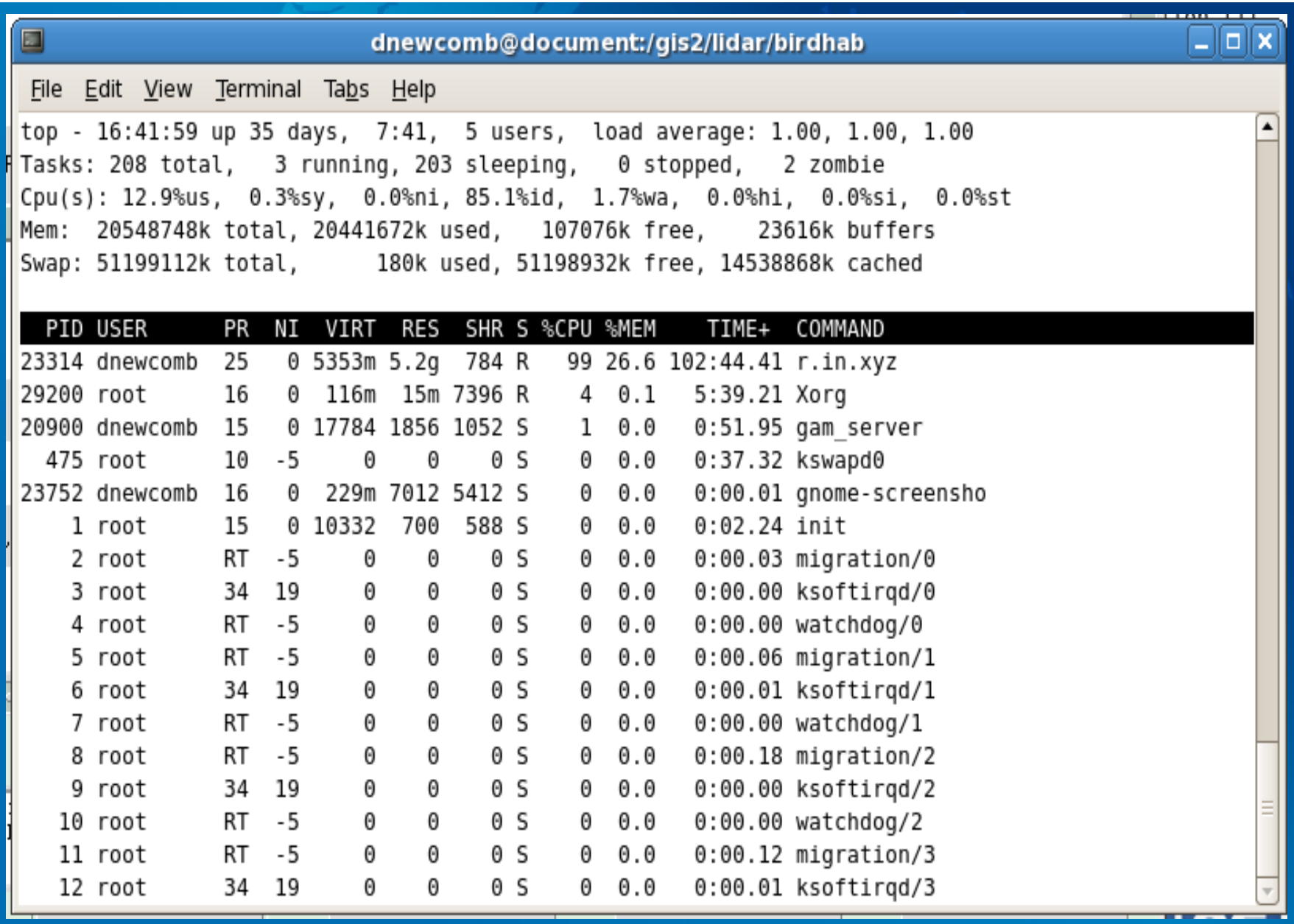
GRASS is single-threaded, each process limited to one core. On multi-core systems, can run 3 analysis simultaneously on the same input file.

But the Server was just sitting there...

Had to give test machine back, but new document repository server came in: 2x Quad Core CPUs, 20 GB RAM, 8x 1 TB drives.

Data moved over to document repository server for final processing.

Processing a 379 GB file



Final Input Dataset Sizes

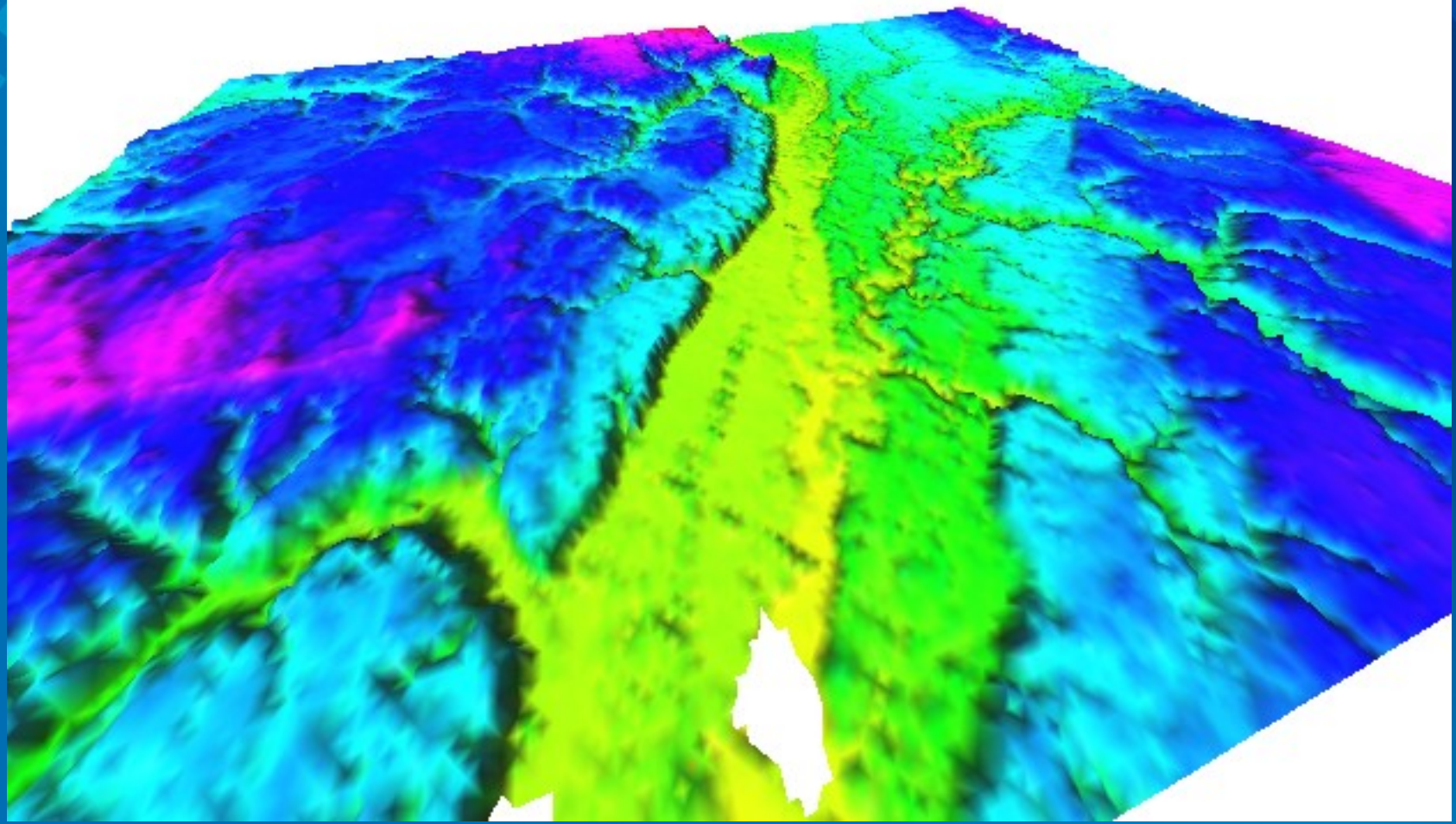
Phase I Contractor A - 92 GB (3.5 hrs)

Phase I Contractor B - 379 GB (Overnight)

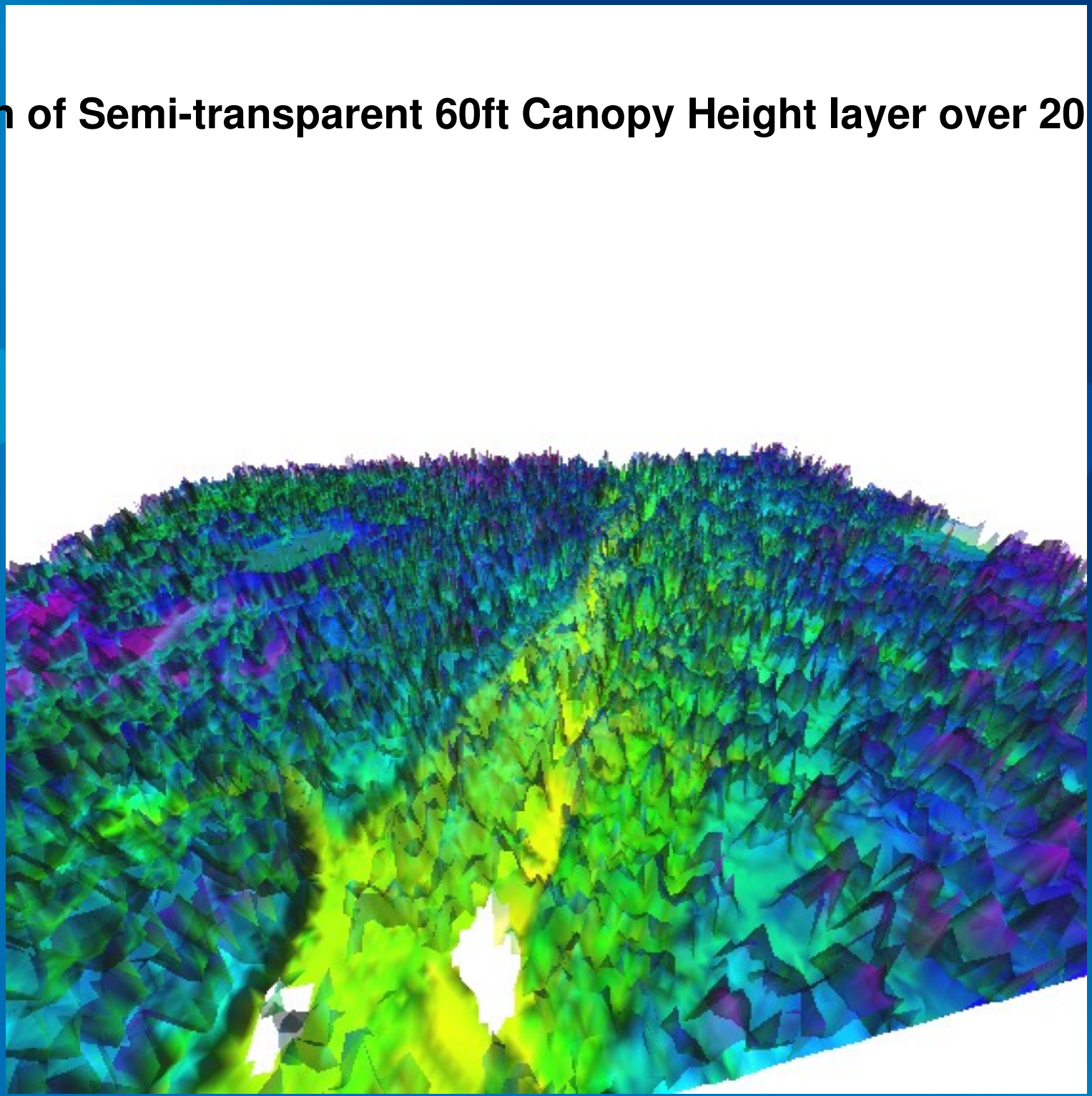
Phase II Contractor A – 150 GB (2 files -4.5 hrs, 1 hr)

Phase III Contractor A – 131 GB (4.5 hrs.)

3D Visualization of 20ft elevation grid



isualization of Semi-transparent 60ft Canopy Height layer over 20 ft elevatio



Openlayers

http://www.openlayers.org

Written in Javascript

Aggregates data from several different sources/ data types

Simple data locally complex data from other sources

Openlayers Example

http://maps.co.mecklenburg.nc.us/gp/

http://maps.co.mecklenburg.nc.us/ft/?p=307

Spatial analysis through the web

- Web – based Mapping tools become more powerful
- Nationwide GAP data analysis project
- http://www5.basic.ncsu.edu/
- http://wiki.openstreetmap.org/wiki/Potlatch/Develop