Development of a Software Framework for Remote Sensing Data Processing

Leland Pierce
March 30, 2015
Overview

What is Remote Sensing?
Why develop a new library?
Development of the new library
What is Remote Sensing?

Airplanes and Satellites imaging the Earth
What is Remote Sensing?

Need to perform a significant amount of processing in order to make the data useful.

Calibration:
What is Remote Sensing?

Geometric Correction:

(Karen Joyce)
What is Remote Sensing?

Land Cover Classification:
What is Remote Sensing?

Mosaicking:
What is Remote Sensing?


(seos-project.eu)
What is Remote Sensing?

There are many other processing functions…

Many tools have been developed over the years to apply these, both commercial and open-source.

I’ve been writing such codes for 25 years.

These need to be integrated in order to make using them easy and efficient.
What is Remote Sensing?

Many Tools: ASF-MapReady, Beam, Cimes, ClasLite, Doris, ESMF, FDO, Fmaps, Fusion, FWTools, GCTP, GDAL/OGR, GDL, GEOS, GeoServer, GeoTools, GeoTrans, GeoPy, GMT, GRASS, QGIS, GSLIB, GStat, GSTL, GvSig, HDF5, ILWIS, ImageMagick, iNVT, ITK, LAS, MapServer, MicMac, Moves, MRS, NCAR-graphics, NCL, NEST, NetCDF, OGDI, OpenCV, OpenEV, OpenLayers, OpenStreetMap, Opticks, OrfeoToolbox, OSSIM, PolSarPro, PostGIS, Proj4, ProSAIL, RadarsatLib, RandomForests, RAT, R-Landsat, ROIPAC, SAGA, SDTS, SGEMS, Shapely, SpatialLite, SWAP, Swarm, uDig, VTP, WAIR, ArcGIS, PCI, ....and many others....
What is Remote Sensing?

QGIS:
Why Develop a new Library?

Lots of libraries and GUIs available

BUT:

Using them requires a huge effort in transporting data between the different codes.

SO:

Want to integrate these into a seamless tool
Why Develop a new Library?

Many tools:
...can’t handle large datasets (>2GB)
...can’t process in parallel
...can’t read in the dataset(s) you need
...don’t include the tool(s) you need
...aren’t cross-platform
...are proprietary: can’t be extended, etc
Development of the new library

Overall Architecture:
1. library (C/C++)
2. command-line (Python)
3. cross-platform GUI (Python)
4. web tools
5. documentation, education, training
Development of the new library

Web Tools

Command Line (Python)

GUI (Python)

library (C/C++)

Documentation, Education, Training
Development of the new library

Choice of Programming Language:

- must be compiled, for speed
- C, C++, Fortran, others?
  - use all, but mostly C++
- C++ only recently added stuff that’s important for making numerical programs straightforward & fast:
  - copy and move of huge objects
  - efficient iteration
  - language features to make this easy
Development of the new library

1. Library Components:
   1.1. file format and I/O
   1.2. data and metadata formats
   1.3. pre-processing tools, filters, …
   1.4. analysis tools
   1.5. higher-level science application tools
Development of the new library

Science Applications

analysis tools: calibration, rectification...

readers, writers, pre-processing, filters, projections, modeling, classification

Data File: data formats and metadata
Development of the new library

1.1 File format and I/O:
- need a standard format, and library
- low-overhead, fast access
- parallel access
- many data-types and data-formats
- flexible metadata

...use HDF5
Development of the new library

1.2 Data and metadata formats:
- bits, bytes, integers, reals, complex
- n-dimensional arrays
- efficient access to rectangular subset
- named metadata items, all types, dims
- directory structure, with named items

...use HDF5
Development of the new library

1.3 Pre-processing tools and filters: Objects

- First implement standard objects:
  - rasters, vectors, meshes
- efficient both for storage and for processing
- intuitive so users can easily use them
- follow standards (data and metadata)
Development of the new library

1.3 Pre-processing tools and filters: Images

- Each image is made from 1 or more rasters
- Each raster is a single scalar per pixel
- Number of pixels and sizes of pixels can vary between rasters
- An image can contain rasters of different types

This is quite different from most image libraries.
Development of the new library

1.3 Pre-processing tools and filters: Vectors

● Store as relational database of geographic coordinates
● points, lines, polygons
● each item has arbitrary attributes describing it, like: name, class, area, perimeter, population, disease incidence, etc (any number, any type)

A spatial-relational database enables efficient spatial searching: use spatialLite
Development of the new library

1.3 Pre-processing tools and filters: Readers

- There are 100s of sensors, with 1000s of data formats
- Must support reading ALL of these
- data/metadata format is needed for each “kind” of data, for sensor-agnostic processing tools
- major “kinds” are radar, optical, lidar, etc
- each “kind” has sub-kinds, but not too many

...use available standards, create new ones as needed
Development of the new library

1.3 Pre-processing tools and filters: Readers

- Huge and never-ending task, as new sensors appear every year.
- Using open-source tools, modification of one reader to create a new one makes this more manageable.
- Existing reader codes often avoid the difficult part of creating standard metadata

...Use ISO metadata standard, Military standards, and standards from professional societies
Development of the new library

XML Metadata Example:

```xml
<gmi:MI_Metadata...>
<gmd:contact xlink:title="Stanford Geospatial Center">
  <gmd:CI_ResponsibleParty uuid="09A95C420FB821476665893256MOME37">
    <gmd:organisationName>
      Stanford Geospatial Center
    </gmd:organisationName>
    <gmd:contactInfo xlink:type="simple">
      <gmd:CI_Contact>
        <gmd:address xlink:type="simple">
          <gmd:CI_Address>
            <gmd:deliveryPoint>Branner Earth Sciences Library</gmd:deliveryPoint>
            <gmd:deliveryPoint>397 Panama Mall</gmd:deliveryPoint>
            <gmd:city>Stanford</gmd:city>
            <gmd:administrativeArea>California</gmd:administrativeArea>
          </gmd:CI_Address>
        </gmd:address>
      </gmd:CI_Contact>
    </gmd:contactInfo>
  </gmd:CI_ResponsibleParty>
</gmd:contact>
</gmi:MI_Metadata>
```
Development of the new library

1.3 Pre-processing tools and filters: Writers

- Choose several standard formats for data output
  - arcgis, pci, grass, envi, ...
  - geotiff, NTIF, jpeg, gif, ...
- Enables interoperability with other software
- This includes using non-standard metadata formats as needed

...The metadata requirements are daunting
Development of the new library

1.3 Pre-processing tools and filters: Maps

- Maps have coordinates and projections
- Coordinates are based on the Earth model used
- Image and Vector data must be interconvertible between all possible combinations of Earth models, projections, coordinates

...Use Proj4 library

(progonos.com)
Development of the new library

1.3 Pre-processing tools and filters: Maps

- An example of how important this is:
- Got lat/long from a USGS map to determine corner coordinates of an image
- Compared with lat/long in the image metadata: 10 miles off!
- Because the image and map used different datums (models of the Earth)
Development of the new library

1.4 Analysis tools:

Calibration, rectification, resampling, box-car averaging, edge-enhancement, filtering, conversion between raster & vector, texture, model decompositions, interferometry, NDVI, pansharpening, cloud detection, fractional cover, land-cover classification, Leaf-Area Index, statistical models, etc.

...100s of these, each an implementation of a method from a journal article
Development of the new library

1.4 Analysis tools:

Simulations of remote-sensing instruments, given a description of the landscape: optical, radar, etc.

Requires detailed 3D/material models of trees, etc.

Finally, we want to “invert” the measurements to estimate the landscape parameters: many methods, many from AI.

...again: hundreds of these, each an implementation of a method from a journal article.
Development of the new library

1.5 Higher-level science application tools:

- soil moisture/roughness,
- forest height/ biomass, terrain height,
- land subsidence, logging detection,
- glacier motion/height, wetland modeling/management, etc.

...again: 100s of these, each an implementation of a method from a journal article
Development of the new library

1.5 Higher-level science application tools:
Example: wetland management
Use elevation, soil type, veg maps to model water flow, veg habitat.
Include model of plant response to herbicides to predict effects, then validate with remotely-sensed veg maps over time.
Development of the new library

2. Command-line

- Use an existing tool, with many libraries
- Use a popular tool, because:
  - users already know it
  - likely to be supported for a while
- Use a cross-platform tool
- Use a tool with an integrated GUI library

...choose Python
Development of the new library

3. Cross-Platform GUI

- Python comes with several GUI toolkits
  - Tkinter
  - wxWidgets
  - Gtk
  - Qt
  - ...and others
- Currently using Tkinter, but could change...
Development of the new library

3. Cross-Platform GUI

Python GUI using TKinter: copied a commercial GUI
Development of the new library

4. Web tools

- tool enabling same GUI as on desktop
- google-earth-like tools
- tools like city property/maps
- tools like magic-bus
- tools like airplane flight-paths

...yet to do any serious work on this
Development of the new library

5. Documentation, education, training

- without this, all the rest will never be used
  - develop different materials for different users:
    - developers
    - expert users (using C++ or Python)
    - novice users, application-oriented
  - different kinds: written, web-based, videos, webinars, in-person classes
  - goes beyond “this function does this”
  - must include how to put tools together to solve real-world problems

...have only written some of the docs so far
Summary

First version: 700,000 lines of C; 500,000 lines of Python. (at 10-20% of full functionality)

Next version:

- C++ with shallow object hierarchies (easier to learn/modify for non-experts)
- GUI integrated with command-line
- GUI easily user-extensible
- Find/implement more metadata standards
- Get students to help
Questions?

online: www.eecs.umich.edu/~lep/geoscilib2015.pdf