Towards Open Source Remote Sensing Software – A Survey

Yinghui Zhao[1,2]

[1] Department of Water Resources,
Zhejiang Tongji Vocational College of Science and Technology,
Hangzhou, 311231, China
[2] School of Earth Sciences and Engineering, Hohai University,
Nanjing, 210098, China
zhaoyinghuihust@gmail.com

Abstract. Remote sensing is one of the best ways for earth observation and environment monitoring due to its spatial and temporal capability for long term and large scale regions. Typical remote sensing software consists of modules including image preprocessing, pixel manipulation, complicated calculation and transformation, interactions with other GIS/RS software, etc. Currently, open source remote sensing is emerging as a promising solution for commercial, governmental, and scientific applications. In this paper we review the state of some open source remote sensing software and make a comprehensive comparison on their general functionalities.

Keywords: remote sensing, open source, spectral image processing.

1 Introduction

Remote sensing is the acquisition of information about an object or phenomenon on Earth (on the surface, in the atmosphere or oceans) from a distance, typically from aircrafts or satellites [1-4]. Passive remote sensing and active remote sensing are main types of remote sensing, where passive sensors detect natural radiation that is emitted or reflected by the targeted object or surrounding areas, especially based on the reflected sunlight of radiation. Typical examples of passive remote sensors include film photography, infrared, charge-coupled devices (CCDs), and radiometers. Since passive remote sensing records radiation that is reflected from Earth's surface, usually from the sun, passive sensors can only be used to collect data during daylight hours. In contrary, active remote sensing uses sensors to emit energy to scan targeted objects and areas then detects and measures the radiation that is reflected or backscattered from the targets. A typical laser based sensing system projects a laser onto the surface of the target (usually the Earth) and then measures the time delay that it takes for the laser to reflect back to its sensor (the laser source). Typical active remote sensing examples include RADAR and LiDAR where the time delay between emission and return is measured, establishing the location, speed and direction of the targeted object [5]. Due to its non-contact nature, remote sensing is widely used in various

F. Bian et al. (Eds.): GRMSE 2013, Part I, CCIS 398, pp. 336–347, 2013.

fields, especially in conditions where it's dangerous or inaccessible to collect data on targeted areas, for an instance, earth observation. Moreover, remote sensing is the best way to monitor very large scale areas in variable spatial and temporal resolutions. For example, remote sensing is the best way to monitor coastal deforestation, glacial features in Arctic and Antarctic regions, global climate change, land use and transformation, hazard assessment, natural resource management, etc.[6,7]. Currently, remote sensing has changed, even replaced the costly and slow ways of data acquisition in many fields, including meteorology, ecology, hydrology, agriculture, military, city planning, public health, archaeological investigations, disease mapping, military observation and geomorphological surveying, etc.

By satellite, aircraft, spacecraft, ship, and helicopter, data is acquired and created automatically from the sensed target by means of various kinds of acquisition systems. The acquired satellite and aerial images are geospatial data and contain more raw data and information than can be easily dealt with, for example, for direct visualization and human interpretation. The information explosion of raw data about our earth does us no good if we cannot turn it into understandable information. Therefore, the acquired data, mostly, the images are processed by computer software, sometimes with the help from human [8-12]. GPU based computing cluster is one of the promising paradigm for large scale remote sensing image mining and analysis with better performance-price ratio [13-17].

A substantial challenge and considerable efforts have been made over the past 4 decades to develop remote sensing related techniques. With the advances in sensor technologies, the spatial resolution(the size of a pixel that is recorded in a raster image – typically pixels may correspond to square areas ranging in side length, for example, 1 pixel stands for a 30 meters by 30 meters area), spectral resolution(the wavelength width of the different frequency bands recorded – usually, this is related to the number of frequency bands recorded by the remote sensing platform and specific devices), radiometric resolution(the number of different intensities of radiation the sensor is able to distinguish), and the temporal resolution are greatly improved. Hence the size of sensed image is becoming larger and larger. They also demand tremendous computing power for sensed image processing and complicated analysis. For an instance, large-scale remote sensing observation and data analysis system requires technique support of high performance computing due to the processing and storage requirements, especially parallel processing techniques and massive data storage and management.

Remote sensing software is almost developed simultaneously with the advances in remote sensing technologies and computer science. The remote sensing software has been developed for decades and commercial software packages have been successfully developed, such as ERDAS Imagine [18], ENVI [19], ER Mapper [20], PCI Geomatic [21], etc. They provide powerful functionalities for calculation and analysis, including mapping, imaginary, tracking, and complicated fusion analysis.

Meanwhile, in the last several decades the open source software movement has received enormous attention from the academia and industry [22-25]. It is often regarded as a fundamentally new way to develop high quality software and it also poses a serious challenge to the conventional commercial software businesses that dominate most software markets today, sometimes even the whole software ecosystem. Nowadays, in the software ecosystem, open source software plays an important role as alternatives for commercial software. From the cost point of view, the open source software development enables the development of software algorithms without spending thousands of dollars for commercial applications. Moreover, open source software development speeds up the development time, produces high-quality codes and allows for the maintenance and distribution of the software to be completed by users from around the world on the internet. In fact, academic institutions, government organizations, entrepreneurial developers and the business communities have benefited a lot from open source software.

Open source remote sensing software is one of the reflection of the open source software movement where many open source remote sensing software are as high as quality with commercial software [26-29].For example, the open source GeoTIFF [30] project is the leading commercial standard for geospatial tiff files adopted by most of the existing commercial remote sensing and GIS software.

The rest of the paper is organized as follows: In Section 2, we present the introductions of some selected open source remote sensing software. We made a comprehensive comparison on the selected open source remote sensing software in Section 3. Finally, we summarize the paper in Section 4.

2 Examples of Selected Open Source Remote Sensing Software and Programming Libraries

In this section, we will select some open source remote sensing software for introduction and comparison. Thanks to the advances in computer programming, all these software evolves after their initial version, some of them even for decades. Most of them are object or component oriented software [31-33].

2.1 OSSIM

OSSIM(Open Source Software Image Map) [34]is a library set that provides high performance remote sensing, mapping and image processing capabilities[35]. OSSIM provides the capability to process images from satellite and aerial cameras and transform them into accurate image maps associated with three-dimensional positions on the Earth. Usually the remote sensing instruments are capable of capturing spectral information from the electromagnetic spectrum outside of human visual range. OSSIM allows this information to be processed for a wide variety of applications, including precision farming, environmental assessment and urban planning. The library is capable of concurrently handling many large image files of arbitrary pixel depth. It has been designed from the ground up to support high performance parallel processing, map projections, three-dimensional models and scientific applications. The library can serve as the basis for a number of significant applications and tools in the remote sensing and geographical information systems areas. The OSSIM library is implemented in C++ language and provides bindings to several libraries, including wxWindows[36], GDAL[37] and GRASS[38]. wxWindows is a framework that

provides a generic GUI(graphical user interface) interface for programmers. The resulting code can then be targeted to Windows, UNIX and Mac platforms respectively. Python and XML support are also provided in the OSSIM library. For example, OSSIM viewer can display a wide range of imagery and can also map data sets, roam, zoom, pan and track the latitude and longitude coordinates of the cursor.
OSSIM has been designed to support dynamic linking of image chains. Images can be opened in their native format and sampled on demand to fill a viewport on the screen or to generate an output product. The dynamic linking allows for reprojections, filters, combiners and other image processing functions to be added or removed easily to produce the desired result. Multiple image chains can be linked together to provide more complex analysis products and views such as mosaics and fusions. OSSIM is also designed to provide nondestructive processing, parameter driven options and a state-management system that allow for taking snapshots of a work in progress and restoring them during later sessions. Functions from the library can be linked together dynamically and combined with other image chains to create the desired product or view. The end of those chains can be combined to create composite processing products. OSSIM can also attach imagery and maps in their native map projections, window into a geographic area of interest (AOI) and resample to common scales and orientations [34].

2.2 GRASS

GRASS (Geographic Resources Analysis Support System)[38], is Geographic Information System and Remote Sensing software used for data management, image processing, graphics production, spatial modeling, and visualization of many types of data. GRASS contains over 350 modules to render maps and images on monitor and paper. It also provides functions such as: manipulate raster and vector data including vector networks; process multispectral image data; and create, manage, and store spatial data. GRASS also offers both an intuitive GUI as well as command line syntax for ease of operations. GRASS GIS can interface with printers, plotters, digitizers, and databases to develop new data as well as manage existing data[38].

In GRASS, imagery data is treated as raster maps but there are a number of special modules specifically designed to help with image processing. GRASS has tremendous capabilities, including raster analysis,3D-Raster (voxel) analysis, vector analysis, point data analysis, image processing, DTM-analysis, geocoding, visualization, map creation, SQL-support, geostatistics, erosion modeling, landscape structure analysis, solution transport, watershed analysis, etc.

2.3 ORFEO Toolbox

The vision of the Orfeo Toolbox (OTB)[39] is a set of algorithmic components, adapted to large remote sensing images, which allow to capitalize the methodological know how, and therefore use an incremental approach to benefit from the results of the methodological research. OTB is distributed as an open source library of image processing algorithms. In OTB, all the details of all the algorithms are open source. OTB is based on the medical image processing library ITK and offers particular functionalities for remote sensing image processing in general and for high spatial resolution images in particular. Targeted algorithms for high resolution optical images (SPOT, Quickbird, Worldview, Landsat, Ikonos), hyperspectral sensors (Hyperion) or SAR (TerraSarX, ERS, Palsar) are available. Most functionalities are also adapted to process huge images without the need for a supercomputer by using streaming and multi-threading as often as possible. Moreover, OTB provides a number of heavily documented functionalities, such as image access (optimized read/write access for most of remote sensing image formats, meta-data access, and visualization), standard remote sensing preprocessing, filtering, feature extraction, image segmentation, classification, change detection, information extraction for integration in GIS and mapping systems [39].

2.4 Opticks

Opticks[40] is an expandable remote sensing and imagery analysis software platform that supports imagery, video (motion imagery), Synthetic Aperture Radar (SAR), multi-spectral, hyper-spectral, and other types of remote sensing data. Opticks supports processing remote sensing video in the same manner as it supports imagery, which differentiates it from other remote sensing applications. Opticks can also be used as a remote sensing software development framework. Developers can extend Opticks functionality using its plug-in architecture and public APIs.

Opticks can also be used as a software development framework. The Opticks community provides and supports a public SDK which includes a documented API as well as several extension tutorials. The Opticks website hosts a variety of extensions, some of which are developed and maintained by the same development team as Opticks.

Opticks supports more than 20 image formats, provides manipulations like zoom, pan, rotate of spatially large datasets and huge images, has quickly layer GIS features, annotations, results, and other information ,image display controls such as colormap, histogram, transparency, etc. It also supports for datasets larger than 4GB and combines steps using graphical wizards, processing data in its native interleave of BIP, BSQ or BIL and various extensions to drop in new capability.

2.5 LibGeoTIFF

TIFF format is the most popular and versatile raster data format in the world today. TIFF is an industry class format for storage, transfer, display, and printing of raster images, such as clipart, logotypes, and scanned documents. Today, TIFF is being used for storage of map information, too.

GeoTIFF[30] is a public domain metadata standard which allows georeferencing information to be embedded within a TIFF file. It’s a TIFF based interchange format for georeferenced raster imagery. The potential additional information includes map projection, coordinate systems, ellipsoids, data, and everything else necessary to establish the exact spatial reference for the file. The GeoTIFF format is fully compliant with TIFF 6.0 specification, so software incapable of reading and interpreting the specialized metadata will still be able to open a GeoTIFF format file.

The TIFF imagery file format can be used to store and transfer digital satellite imagery, scanned aerial photos, elevation models, scanned maps or the results of many types of geographic analysis products. Over the past several years many users of such images have urged geographic data suppliers to provide imagery in TIFF format. TIFF is the only full-featured format in the public domain, capable of supporting compression, tiling, and extension to include geographic metadata. GeoTIFF implements the geographic metadata formally, using compliant TIFF tags and structures.

GeoTIFF refers to TIFF files which have geographic (or cartographic) data embedded as tags within the TIFF file. The geographic data can then be used to position the image in the correct location and geometry on the screen of a geographic information display. GeoTIFF is a metadata format, which provides geographic information to associate with the image data. But the TIFF file structure allows both the metadata and the image data to be encoded into the same file. GeoTIFF makes use of a public tag structure which is platform interoperable between any and all GeoTIFF-savvy readers. Any GIS, CAD, Image Processing, Desktop Mapping and any other types of systems using geographic images can read any GeoTIFF files created on any system according to the GeoTIFF specification [30].

Libgeotiff is an open source library normally hosted on top of libtiff for reading, and writing GeoTIFF information tags.

2.6 GDAL

Image manipulation libraries play a crucial role in all the remote sensing software and they provide base functions for pixel-level image manipulation and calculation. GDAL [37] is the geospatial data access library that provides read/write access to a large number of geospatial data formats including raster and vector data. GDAL is a translator library for raster geospatial data formats and provides a single abstract data model to the calling application for all supported formats. Within the GDAL source tree, there is another library, i.e., the OGR library which provides a similar capability for vector data. The GDAL/OGR libraries are backend supporting libraries in many remote sensing and GIS software including ArcGIS[41], FWTools[42], GRASS GIS[38], MapServer[43], OSSIM[34], OpenEV[44], Orfeo toolbox(also known as OTB)[39], Quantum GIS(i.e., QGIS)[45],etc.

In GDAL, the data model is the type of information that a GDAL data store can contain, and their semantics. A dataset (represented by the GDAL Dataset class) is an assembly of related raster bands and some information common to them all. In particular the dataset has a concept of the raster size (in pixels and lines) that applies to all the bands [37]. The dataset is also responsible for the georeferencing transform and coordinate system definition of all bands. The dataset itself can also have associated metadata, a list of name/value pairs in string form. The GDAL dataset, and raster band data model is loosely based on the OpenGIS Grid coverage specification.

2.7 GEOTRANS

The National Imagery and Mapping Agency (NIMA) GEOTRANS(Geographic Translator)[46] package provides a valuable tool chest of certified map projection transformations, file formats and useful functionality that has been subsequently modernized to object-oriented code. GEOTRANS provide function to convert geographic coordinates among a wide variety of coordinate systems, map projections, and data. To convert a set of coordinates, simply select the coordinate system or map projection, the datum, in which the coordinates are defined, then enter the source coordinates, select the coordinate system or map projection, and the datum, to which the user want the coordinates to be converted, and then can start the convert procedure. Currently, there are twenty-five different coordinate systems, map projections, grids, and coding schemes, and over two hundred different datums that are supported.

GEOTRANS can also be used to efficiently convert large numbers of coordinates contained in either text files (.txt) or comma separated values (.csv). A multiline file header defines the coordinate system or map projection, and datum, of the coordinates contained in the file. Following the header, each line contains a single set of coordinates, separated by commas. Using the GEOTRANS file processing interface, the user can select an existing file of coordinates to be converted, define the coordinate system or map projection, and the datum, to which the user want to convert the coordinates, and specify the name and location of the output file that is to be created. GEOTRANS then converts the input file and creates the output file as a single operation.

3 Comparison of Selected Open Source Remote Sensing Software and Programming Libraries

Table 1 and table 2 compare the open source remote sensing software and libraries surveyed in this paper including various attributes.

Table 1. Comprehensive comparison for Open Source Remote Sensing Software

Software	Language	License	Support Platform	Library	Parallel Processing	Features	GUI/CLI	Plug-in	Image Support	Image Format	Backend Institution/ Developer
OPTICKS	C++	LGPL	W/S	N/A	N/A	ABCD	GUI/CLI	Yes	ABC	A	NASIC
OTB	C++	CeCILL	U/L/W/M/S	N/A	N/A	ABCD	GUI/CLI	Yes	ABC	A	CNES
OSSIM	C++	LGPL	L/W/M	GDAL	Yes	ABCD	GUI/CLI	Yes	ABC	A	OSGeo
NEST[47]	Java	GPL	L/W/M	N/A	Yes	ABCD	GUI/CLI	Yes	C	AB	ESA
BEAM[48]	Java	GPL	L/W/M	N/A	Yes	ABCD	GUI/CLI	Yes	C	AB	ESA
GRASS	C	GPL	L/W/M	GDAL	Yes	ABCD	GUI/CLI	Yes	ABC	AB	CERL/ OSGeo
QGIS	C++	GPL	L/U/M/W/A	GDAL	N/A	ABCD	GUI/CLI	Yes	ABC	A	OSGeo
OpenEV	Python	LGPL	L/W/S/I	GDAL	N/A	N/A	CLI	Yes	ABC	A	CGDI
Parbat[49]	Java	N/A	L/W/S	GDAL	N/A	A	GUI/CLI	No	A	A	A. Lucieer
PolSARpro[50]	C	GPL	L/U/M/W/S	ActiveTcl	N/A	A	GUI/CLI	Yes	SAR	A	ESA
ILWIS Open[51]	C++	GPL	L/U/M/W/S	N/A	N/A	ABCD	GUI/CLI	Yes	ABC	AB	ITC

Table 2. Comprehensive comparison for Open Source Remote Sensing Libraries

Libraries	Language	Binding language	License	Support Platform	Parallel Processing	GUI /CLI	Plug-in Extension	Image Source Support	Backend Institution Developer
GDAL	C++	C/C++/Perl/Python/VB6 /Ruby/Java/C#/R	X/MIT	U/ L/S/M/W	Yes	CLI	Yes	N/A	OSGeo
libgeotiff	C	C/Java	X/MIT	U/ L/S/M/W	N/A	N/A	Yes	GeoTIFF	OSGeo
GCTP	C	Fortran/C	N/A	U/ L/W	N/A	N/A	N/A	N/A	NASA
Proj/PROJ4[52]	C	C	MIT	U/ L/W	N/A	CLI	N/A	N/A	USGS
GeoTools[53]	Java	Java	LGPL	U/ L/W	N/A	GUI /CLI	N/A	NA/	OSGeo
LASlib[54]/ LASzip[55]	C++	C++	LGPL	U/ L/W	N/A	CLI	N/A	LAS	rapidlasso
libLAS[56]	C/C++	Python/C++/C#/VB.Net /Ruby	BSD	U/ L/W/M	N/A	CLI	N/A	LAS	IGSB
GeoTrans	C++	C/C++/Python	N/A	U/ W	N/A	CLI	N/A	N/A	NIMA

Note:

1. Licensing marked N/A means fully free for public domain and not dedicated to a specific license like GPL or LGPL or X/MIT. GPL means version greater than 2.0 except specified explicitly.

2. We define the features of the software as three categories as follows:

Type A: General features and functions for remote sensing image processing, including:

(1)Image manipulation: zoom, pan, rotate, metadata access, image visualization;
(2)sensor geometry: sensor models, cartographic projections;
(3)radiometry: atmospheric corrections, vegetation indices;
(4)filtering: blurring, denoising, enhancement;
(5)fusion: image pansharpening;
(6)feature extraction: interest points, alignments, lines;
(7)image segmentation: region growing, watershed, level sets;
(8)classification: K-means, SVM, Markov random fields;
(9)change detection.
(10)object based image analysis.
(11)geospatial analysis.

Type B: GIS support, GIS layer constructions, annotations,
Type C: Special functions, in memory and on disk processing, Batch processing
Type D: Scripting,IDL,Python

3. We classify the images source used in the software or libraries into three categories as follows:

Type A: optical images
Type B: hyperspectral sensors
Type C: SAR

4. We classify the images formats used in the software or libraries into three categories as follows:

Type A: NITF2.0/2.1, ASPAM/PAR, CGM, DTED, ENVI, Generic RAW, ESRI Shapefile, HDF5, AVI, MPEG, JPEG, GIF, PNG, BMP, TIFF, GeoTIFF
Type B: NetCDF, CEOS
Type C: others

5. In Support platform, W is Windows, U is Unix, L is Linux, S is Solaris, M is MacOS,A is Android,I is IRIX

4 Conclusion and Future Work

The open source software challenge is often described as much more fundamental, and goes to the basic motivations, economics, market structure, and philosophy of the institutions that develop, market, and use software. In the remote sensing area, the open source remote sensing software plays an important role for commercial, personal and scientific applications. In this paper we review the state of some open source remote sensing software and make a comprehensive comparison on their general functionalities.

In this paper we do not compare the selected software exhaustively but we compare some functionality aspects that are import for software selection and implementation in real remote sensing system development. We also present some discussions for open source software selection in the following except for the comparison in the above sections.

(1) Parallel processing support is very import for remote sensing software. Large scale remote sensing image processing, especially for the high resolution images is highly computing-intensive and time-consuming and it poses serve computational challenges for modern computing systems. The inherent nature of raster data makes it very suitable for parallel processing of large number of images. However, different domain specific applications demands different processing methods and analysis. For example, different applications demand increasing amounts of spatial, temporal, and spectral resolution images for different reception, archival, cataloging, user query, processing, manipulation, and analysis of the remote sensing image data. Currently there is no standard high performance parallel remote sensing image processing specifications or implementation references. Usually the researchers and developers use standard parallel computing techniques to accomplish parallel performance and speedups. Multi-threaded image processing is often used in real applications by using of super computers, clusters, and multi-core PC servers[57-59]. However, how to tune the performance of the target high performance computing infrastructure for remote sensing image processing should be addressed further[60,61].GRASS is one of the open source remote sensing software that has capabilities of parallel processing for remote sensing images[62,63].

(2) Scripting and customized development support is another consideration for real world remote sensing applications. Support for scripting language including JavaScript, Python and Perl is essential for customized development and image processing in various environments.

Acknowledgments. The funding support of this work by Natural Science Fund of Zhejiang Province (ZJNSF, Grant No. LY12E09009), Science and Technology Planning Project in Water Resources of Zhejiang Province, China (Grant No. RC1218), and 2012 Special Funding Program in Water Resources of Zhejiang Provincial Departments, are greatly appreciated.

References

1. Davis, S.M., Landgrebe, D.A., Phillips, T.L.: Remote Sensing: the Quantitative Approach. McGraw Press, New York (1978)
2. Goetz, A.F.H., Vane, G., Solomon, J.E., Rock, B.N.: Imaging Spectrometry for Earth Remote Sensing. Science 228, 1147–1153 (1985)
3. Pohl, C., Van Genderen, J.L.: Multisensor Image Fusion in Remote Sensing: Concepts, Methods and Applications. International Journal of Remote Sensing (1998)
4. Green, R.O., Eastwood, M.L., Sarture, C.M., Chrien, T.G., Aronsson, M., Chippendale, B.J., Faust, J.A., Pavri, B.E., Chovit, C.J., Solis, M.: Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). Remote Sensing of Environment 65(3), 227–248 (1998)
5. `http://en.wikipedia.org/wiki/Remote_sensing`
6. Ehlers, M., Janowsky, R., Gähler, M.: New Remote Sensing Concepts for Environmental Monitoring. In: The Conference on Remote Sensing for Environmental Monitoring, GIS Applications, and Geology, Toulouse, France, pp. 1–12 (2002)
7. Ren, H., Chang, C.-I.: Automatic Spectral Target Recognition in Hyperspectral Imagery. IEEE Transactions on Aerospace and Electronic Systems 39, 1232–1249 (2003)
8. Jiang, C., Xu, X., Wan, J., Zhang, J., Li, Y.: Java Multi Threaded Based Parallel Remote Sensing Image Interpretation in Desktop Grid. In: 5th Annual ChinaGrid Conference (China Grid 2010), pp. 51–59. IEEE Press, New York (2010)
9. Plaza, A., Valencia, D., Plaza, J., Martinez, P.: Commodity Cluster-Based Parallel Processing of Hyperspectral Imagery. Journal of Parallel Distributed Computing 66, 345–358 (2005)
10. Plaza, A., Chang, C.-I.: High Performance Computing in Remote Sensing. Taylor & Francis, Boca Raton (2007)
11. Plaza, A., Du, Q., Chang, Y.-L., King, R.L.: High Performance Computing for Hyperspectral Remote Sensing. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 528–544 (2011)
12. Plaza, A., Plaza, J., Paz, A., Sanchez, S.: Parallel Hyperspectral Image and Signal Processing. IEEE Signal Processing Magazine 28, 119–126 (2011)
13. Sanchez, S., Paz, A., Martin, G., Plaza, A.: Parallel Unmixing of Remotely Sensed Hyperspectral Images on Commodity Graphics Processing Units. Concurrency and Computation: Practice and Experience 23, 1538–1557 (2011)
14. Lee, C.A., Gasster, S.D., Plaza, A., Chang, C.-I., Huang, B.: Recent Developments in High Performance Computing for Remote Sensing: A Review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 508–527 (2011)
15. Christophe, E., Michel, J., Inglada, J.: Remote Sensing Processing: From Multicore to GPU. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 643–652 (2011)
16. Mielikainen, J., Huang, B., Huang, A.: GPU-Accelerated Multi-Profile Radiative Transfer Model for the Infrared Atmospheric Sounding Interferometer. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 691–700 (2011)
17. Chang, C.-C., Chang, Y.-L., Huang, M.-Y., Huang, B.: Accelerating Regular LDPC Code Decoders on GPUs. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 653–659 (2011)
18. ERDAS Imagine, `http://geospatial.intergraph.com/`
19. ENVI, `http://www.exelisvis.com`
20. ER Mapper, `http://geospatial.intergraph.com/`
21. PCI Geomatics (2013), `http://www.pcigeomatics.com`

22. McIlhagga, D., Zeiss, G.: Open Source Geospatial An Alternative Business Model for Municipal Governments
23. Dibona, C., Ockman, S., Stone, M.: Open Sources: Voices from the Open Source Revolution. O'Reilly, Sebastopol (1999)
24. Raymond, E.S.: The cathedral and the bazaar (1999), http://www.tuxedo.org/esr/writings/cathedral-bazaar/
25. Vixie, P.: Software Engineering. In: Dibona, C., Ockman, S., Stone, M. (eds.) Open Sources: Voices from the Open Source Revolution, pp. 91–100. O'Reilly, Sebastopol (1999)
26. Christophe, E., Inglada, J.: Open Source Remote Sensing: Increasing the Usability of Cutting-Edge Algorithms. IEEE Geoscience and Remote Sensing Society Newsletter, 9–15 (2009)
27. Steiniger, S., Bocher, E.: An Overview on Current Free and Open Source Desktop GIS Developments. International Journal of Geographical Information Science 23, 1345–1370 (2009)
28. Open Source Initiative, http://www.opensource.org/
29. Paul Ramsey. The state of Open Source GIS.FOSS4G (2007)
30. http://trac.osgeo.org/geotiff/
31. Câmara, G., Souza, R.C.M., Freitas, U.M., Garrido, J.: SPRING: Integrating Remote Sensing and GIS by Object-Oriented Data Modeling. Computers & Graphics 20, 395–403 (1996)
32. Blaschke, T.: Object Based Image Analysis for Remote Ssensing. ISPRS Journal of Photogrammetry and Remote Sensing 65, 2–16 (2010)
33. DeBardeleben, N.A., Ligon, W.B., Stanzione, D.C.: The Component Based Environment for Remote Sensing. In: Proceedings of 2002 IEEE Aerospace Conference, vol. 6, pp. 2661–2670. IEEE Press (January 2002)
34. OSSIM, http://www.ossim.org
35. http://www.remotesensing.org
36. wxWindows, http://www.wxwidgets.org
37. GDAL, GDAL - Geospatial Data Abstraction Library: Version 1.9.2, Open Source Geospatial Foundation (2012), http://gdal.osgeo.org
38. http://grass.osgeo.org/
39. http://www.orfeo-toolbox.org/
40. http://opticks.org/
41. ArcGIS, http://www.esri.com/
42. FWTools, http://fwtools.maptools.org/
43. MapServer, http://mapserver.org/
44. OpenEV, http://openev.sourceforge.net/
45. QGIS, http://www.qgis.org/
46. http://earth-info.nga.mil/GandG/geotrans/
47. The Next ESA SAR Toolbox, http://nest.array.ca/web/nest
48. BEAM, http://www.brockmann-consult.de/cms/web/beam/
49. Parbat, http://parbat.lucieer.net/
50. PolSARpro, http://earth.eo.esa.int/polsarpro/
51. ILWIS Open, http://ilwis.org/
52. Proj/PROJ4, http://trac.osgeo.org/proj/
53. GeoTools, http://www.geotools.org/
54. LASlib, http://www.cs.unc.edu/~isenburg/lastools/
55. LASzip, http://www.laszip.org/

56. libLAS, http://www.liblas.org/
57. Wei, S.-C., Huang, B.: GPU Acceleration of Predictive Partitioned Vector Quantization for Ultraspectral Sounder Data Compression. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 677–682 (2011)
58. Yang, H., Du, Q., Chen, G.: Unsupervised Hyperspectral Band Selection Using Graphics Processing Units. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 660–668 (2011)
59. Goodman, J.A., Kaeli, D., Schaa, D.: Accelerating an Imaging Spectroscopy Algorithm for Submerged Marine Environments Using Graphics Processing Units. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4, 669–676 (2011)
60. Ray, S., Simion, B., Brown, A.D.: Jackpine A Benchmark to Evaluate Spatial Database Performance. In: IEEE 27th International Conference on Data Engineering (ICDE 2011), pp. 1139–1150 (2011)
61. Simion, B., Ray, S., Brown, A.D.: Surveying the Landscape: an In-depth Analysis of Spatial Database Workloads. In: The 20th International Conference on Advances in Geographic Information Systems (SIGSpatial 2012), pp. 376–385 (2012)
62. Osterman, A.: Open Source GIS A GRASS GIS Approach Implementation of the r.cuda.los Module in the Open Source GRASS GIS by Using Parallel Computation on the NVIDIA CUDA Graphic Cards. Elektrotehniski Vestnik 79, 19–24 (2012)
63. Sorokine, A.: Implementation of a Parallel High-Performance Visualization Technique in GRASS GIS. Computers & Geosciences 33, 685–695 (2007)