Korean Journal of Remote Sensing, Vol.26, No.3, 2010, pp.361~371

Open Source Remote Sensing of ORFEO Toolbox and Its Connection to Database of PostGIS with NIX File Importing

Kiwon Lee† and Sanggoo Kang

Dept. of Information Systems Engineering, Hansung University

Abstract : In recent, interests regarding open source software for geo-spatial processing are increasing. Open source remote sensing (OSRS) is regarded as one of the progressing and advanced fields in remote sensing. Nevertheless, analyses or application cases regarding OSRS are not enough for general uses or references. In this study, three kinds of OSRS software in consideration of international popularity, types of functionalities, and development environments are taken into account: OSSIM, Opticks, and ORFEO Toolbox (OTB). First, functional comparison with respect to these is carried out on the level of the preliminary survey. According to this investigation, OTB is chosen as the most applicable OSRS software in this study. Running on OTB, NIX format importing module and database connecting module are implemented for widely general uses and further application. As for an example case, airborne image of NIX format is used to region growing segmentation algorithm in OTB, and then the results are stored and retrieved in PostGIS database to test implemented modules. Conclusively, local customization and algorithm development using OSRS software are necessary to build on-demand applications from the developers' viewpoint.

Key Words : NIX format, Open Source Remote Sensing (OSRS), ORFEO, PostGIS, Segmentation.

1. Introduction

Since the mid-2000s, open source geo-spatial processing software has been regarded as one of important progresses in the geo-spatial communities from both users' side and developers' side, such as OSGeo foundation or FOSS4G (Free and Open Source Software for Geo-spatial) group. Although open source projects have started from a couple of decade, they are affected by new computing paradigm towards openness and sharing of data, information, or processors, as well as various kinds of mashup applications of Google Maps API (Application Programming Interface) in Purvis *et al.*(2006), Park and Lee (2008) and Park and Lee (2009).

As for general information and references of open source geo-spatial software, there are useful overviews and articles such as Hall and Leahy (2008), Benthall (2009) or Markus (2009). Moreover, researches regarding uses of open source in a specific geo-based application domain are also on the

Received June 23, 2010; Revised June 24, 2010; Accepted June 25, 2010.

† Corresponding Author: Kiwon Lee (kilee@hansung.ac.kr)

activating stage such as Steiniger and Bocher (2009), Steiniger and Hay (2009), and Daoyi *et al.* (2010). Nonetheless, these works are focused on open sources for generic GIS dealing with the geo-based objects by vector data models, so that there are rare cases for geo-processing open sources for remote sensing. Thus, the main theme in this work is on open source remote sensing (OSRS) software covering satellite image processing.

Normally, open source software for geo-spatial processing provides dual modes: binary executables for users and full source codes, APIs (Application Programming Interfaces) or development libraries for algorithm developers and programmers. While, it is noted that open source is not same terms with free software, and its opposite is proprietary software, not commercial software. In general, most users who have interests in OSRS software are inclined to compare and assess functionalities of OSRS software with ones of proprietary remote sensing software. But individual OSRS software has its specialty and motivation to fit its purposes, so that simple comparison limited to its providing functionalities may be a trivial detail, especially from application programmers' or algorithm developers' viewpoints.

In this study, three kinds of OSRS software satisfied with certain conditions are considered. As for the conditions, these considerations are used for basic criteria: providing possibly full functionalities including pre-processing and post-processing modules, supporting low level sources to capable of re-compiling or development library or APIs, and no cost for license to modify or edit. After a preliminary investigation on those conditions, three products are selected in this work: OSSIM, Opticks, and ORFEO Toolbox (OTB). These three OSRS software are valuable to implement a certain remote sensing application, and enough to provide unique functionalities or algorithms, but this work is not covered with all features for these.

The purposes of this work are brief comparison regarding development environment and function lists; using one of them, new implementations for Korean geo-spatial image standard format named NIX (National Image Exchange) and open source geo-database management of PostGIS/PostgreSQL are carried out for practical uses for actual applications of segmentation and its results to data base connection.

2. Brief Comparison of Open Source Remote Sensing Software

OSSIM which stands for Open Source Software Image Map is an open source software system for remote sensing, image processing including photogrammetric functions and GIS. Since the mid-1990s, it has been developed as an open source software project, being funded by several US government agencies in the intelligence and defence community. Opticks is open-source software offerings to include new extensions that perform hyper-spectral, multispectral and image spectroscopy analysis by US-based Ball Aerospace and Technologies Corp. released in 2007. It supports traditional imagery and multi-spectral data as well as SAR, hyper-spectral, and motion imagery. Its expansion includes tools to convert imagery from radiance to reflectance, as well as tools to load and display signatures, specifically wavelength versus reflectance. To help grow the practical use of satellite imagery, ORFEO Toolbox (OTB), which stands for Optical and Radar Federated Earth Observation, is OSRS software developed by CNES in France, started from the mid 2000s. OTB is to enable the user to process satellite images from different sources including satellite and image provider with different levels of pre-processing to post-processing, with Radar data processing.

Table 1 is the summary of development environment of these three OSRS software. While, as for users-side, user interfaces of these OSRS software are shown in Fig. 1. Interfaces in OSSIM are composed of three parts: Imagelinker in Fig. 1(A), iview and ossimplant. Opticks provides an integrated user interface with menus for all functions in Fig. 1(B). In OTB, dual access modes are provided: Monteverdi application in Fig. 1(C) and interfaces for modular function.

Fig. 2 represents source components including core classes and extensions and dependency with other open libraries in OSSIM, Opticks, and OTB. This implies that cross- access between them is possible since these three open sources are closely related. In the cases of dependency with other open sources of OTB, GDAL (Geospatial Data Abstraction Library) is a translator library for raster geospatial data formats that is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation. ITK(Insight ToolKit) is for the library for medical or other special image processors including registration and segmentation, and OpenGL and boost are 3D graphical rendering API and graphical presentation,

Table 1. Development environment of OSSIM, Opticks, and OTB

Environment	OSSIM	Opticks	OTB
Operating System	Windows, Linux, Mac, Solaris	Windows, Solaris	Windows, Linux, Mac, Unix
Programming Language	C++, Python	C++, Python, IDL	C++, Python, IDL/ENVI, MatLab, Java
URL	www.ossim.org	Opticks.org	otb.cnes.fr

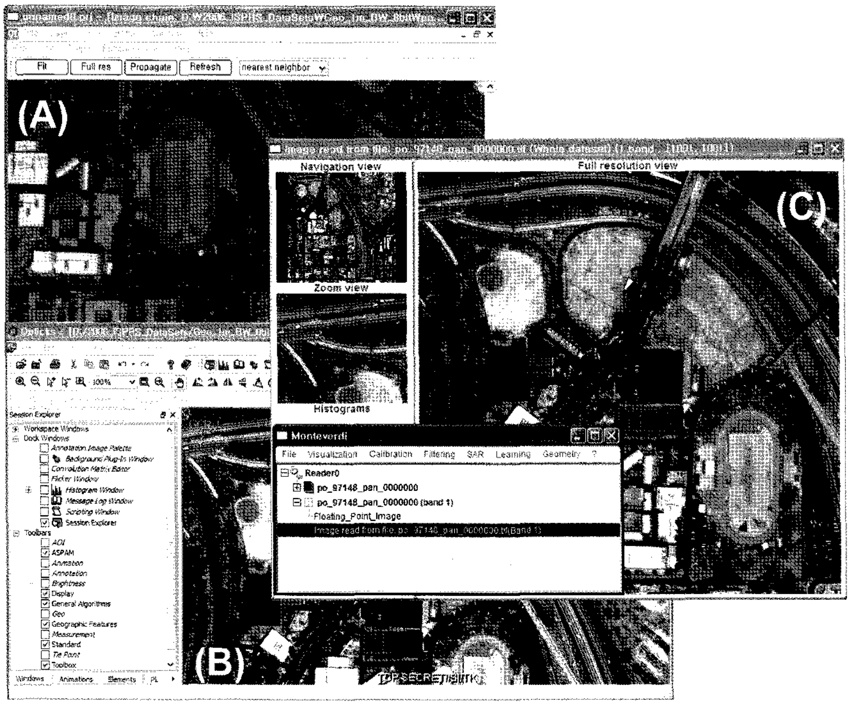

Fig. 1. OSRS software user interfaces: (A) Imagelinker interfaces for basic processing in OSSIM, (B) Integrated user interface of Opticks, and (C) Integrated interface of Monteverdi application in OTB.

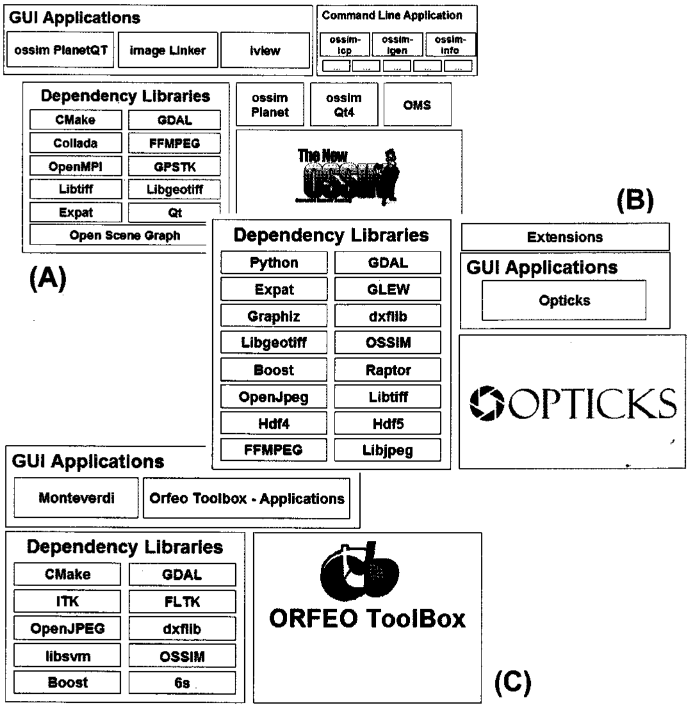

Fig. 2. Open source components and dependencies with other libraries in OSSIM, Opticks, and ORFEO toolbox (OTB).

respectively. More, FLTK, LibSVM, and OpenJPEG are the libraries for graphical user interface, supervised classification of Support Vector Machine algorithm, and Jpeg2000 format, respectively. Further, libKML is for Google vector data format supporting. And 6S is used for radiometric processing. OSSIM is for sensor modelling, DEM modeling and map projections in OTB. Monteverdi in OTB is for uses of GUI to build applications or uses the applications. Table 2 is the comparative list for function types of OSSIM, Opticks, and OTB. This comparison work is basically by tentative results based on users guide book or simple testing, so more detailed investigation regarding applied algorithms and their performance is needed for further analysis. Therefore, the notation of ◐ or ○ is not the symbols representing bad or not good, and these symbols are subject to be changeable by application domain.

According to the previous results in Table 1 and Table 2, and Fig. 2, OTB is chosen as open source to newly implement and test in this work, with the reasons of multi-platform supporting, modularity of user interfaces, and physical connecting with geo-spatial database, besides full functionalities including pre-processing and post-processing. OTB is a somewhat complex system and needs to be validated both at a computing engineering level and at functionality level. It is known that about 3000 C++ classes are already available in the current version of OTB for most of the usual operations on remote sensing images. Besides integrated user interface named Monteverdi for basic processing, the execution of main functions is applicable through each modular interface in OTB applications: fine registration, pireo coarse registration, feature extraction, urban area extraction, road network extraction, object counting, interactive change detection, segmentation, ortho-rectification and

Table 2. Comparative list for function types of OSSIM, Opticks, and OTB. Note: ● (well defined and supported), ◐ (intermediately supported), ○ (weakly supported)

Functions	OSSIM	Opticks	OTB
Band Math	●	●	●
Band Selector	●	◐	●
Atmospheric Correction	●		●
Calibration	◐	○	●
Correction	●	○	●
GCP Editor	◐	●	●
Filtering	●	◐	●
Clustering	◐	●	●
Segmentation	○	○	●
Thresholding	◐	◐	●
Change Detection	◐	○	●
Convolution Matrix	◐	●	●
Feature Extraction	○	○	●
Fusion	●	◐	●
Classification	●		●
HSI Adjustment	●	◐	●
Orthorectification	●		●
Pan-sharpening	◐	◐	●
PCA	◐	●	●
SAR Intensity			●
SAR Despeckle			●
Web Service	●		●
DEM Processing	●	◐	○
DB Supporting	○	◐	◐
Documentation	◐	○	●
Ease of use for general users	◐	●	◐

fusion, radiometric calibration, land cover mapping, supervised classification, image to database registration, and polarimetric synthesis (Cristophe, 2008; Christophe *et al.*, 2008; Guzzonato *et al.*, 2009; Rosario, 2009).

Moreover, OTB supports multi-platform to impose strict design and coding rules thus leading to a robust system less sensitive to particular platform specificities. As for data size, streaming techniques are used to read, process and write the result progressively without having to load the entire image into memory. Streaming capabilities and multithreading, working on clusters where the combination of several processors help to greatly reduce the total processing time, are included in OTB transparently for the user, thus enabling an easy processing of huge images. Recent, algorithm development or actual application studies using OTB are being reported in Christophe and Inglada (2009), Simler *et al.* (2009) and Özdemir *et al.* (2010).

3. Implementations: NIX Importing and PostGIS Connection

Generally, open sources for geo-spatial data processing are very useful resources for application implementation, but mostly they need local customizations. Therefore, for the extended uses of OTB, these two implementations are carried out (Fig. 3 and Table 3): nation-wide standard formats supporting and large database managements with query processor. In Fig. 3, the main engine is OTB, and it provides reader modules for most types of geo-based image formats such as GeoTiff, JPEG 2000, BMP, TIFF, PNG, BSD, BIL, SAR image and other conventional image formats, as well as some vector formats. However, implementation of extension for local standard file format should be done by customizing process.

NIX(National Image eXchange) is the Korean geo-spatial image standard format developed by National Geographic Information Institute. It basically uses JPEG 2000 image compression scheme, tiling, and image pyramid based on wavelet, in code streaming. Fig. 4 summarizes NIX specifications. Though NIX is based on JPEG 2000 compression scheme, but direct reading into OTB is problematic. NIX format decoding and encoding modules are implemented using other open sources such as GDAL and Jasper (Adams, 2006) and a part of codes written for NIX file importing module is shown in Fig. 5.

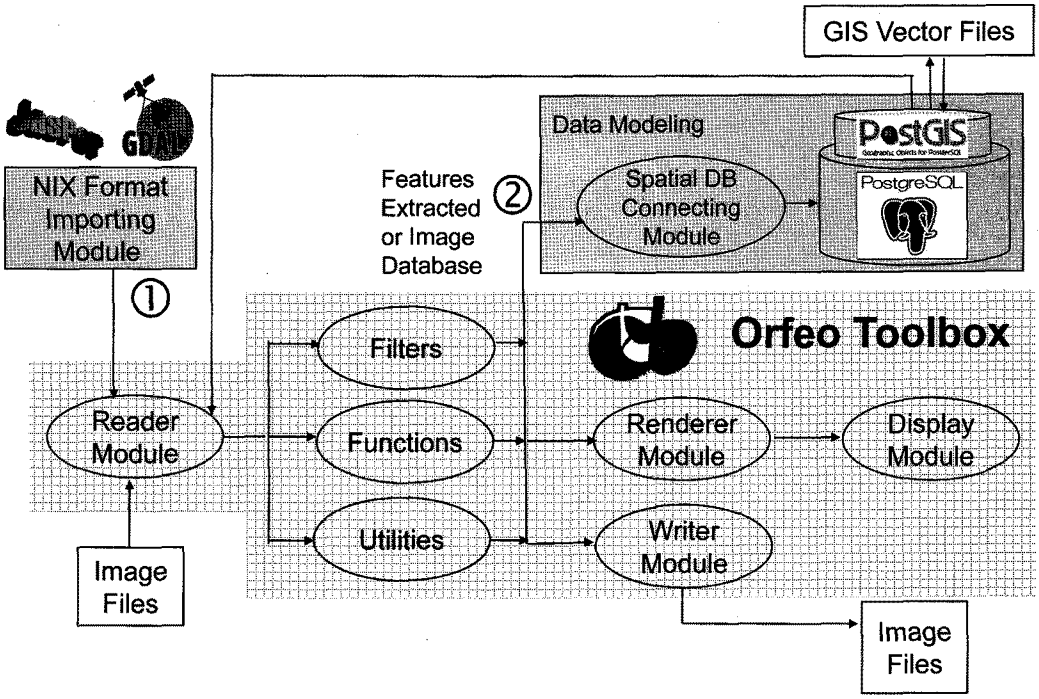

Fig. 3. NIX file importer and PostGIS database connector on OTB, implemented in this study.

Table 3. Open sources versions applied for implementations in this study

Environment	Released Version or Patch
OS	Fedora 12.
OTB Version	3.2
GDAL Version	1.7.2
Jasper Version	1.900.1
PostgreSQL Version	8.4.4
PostGIS Version	1.4.2

As another implementation, database connecting modules are added into OTB. In this work, PostGIS/PostgreSQL, released by Refractions which is Canada-based firm, is considered as the connecting open source database management software. This has very interesting features, paralleled only by the most powerful and expensive proprietary systems; it guarantees the integrity of transactions and supports a wide set of different data types, and it allows the management of very large data bases to hundreds of Terabytes, even with computer of limited power, and allows clustering, load balancing, efficient use of multi-processor computers, and data redundancy. Moreover, it has strong security, based on several

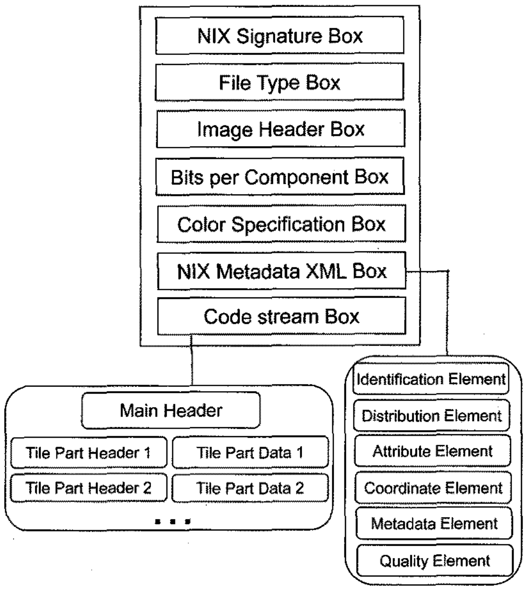

Fig. 4. Specifications for NIX file format, Korean geo-based image standards (http://air.ngii.go.kr/index.do).

```
bool GDALImageIO::CanReadFile(const char* file)
{       ...
        unsigned char ch;    int i = 0, metaDataLength;  unsigned char readByte[4];
        FILE *m_NIX = fopen(lFileNameGdal.c_str(), "rb");
        FILE *nix_metadata = fopen("meta.xml", "w+");
        FILE *m_JPC = fopen("temp.jpc", "wb");
        if(m_NIX == NULL) return false;
        fseek(m_NIX, 89, SEEK_CUR);
        //get metadata length
        while(i<4) {
                readByte[i] = fgetc(m_NIX);
                if(i==3) metaDataLength = byte4ToInt(readByte);
                i++;
        }
        fputs("<?xml version=\"1.0\" encoding=\"utf-8\" ?>", nix_metadata);
        i=0;
        //metadata write
        while(i<metaDataLength-8) {
                ch = fgetc(m_NIX);
                fputc(ch, nix_metadata);
                i++;
        }
        int size = ftell(m_NIX); //jpeg2000 codestream size calculation
        fseek(m_NIX, 01, SEEK_END);
        int size2 = ftell(m_NIX);
        size2 = size2 - size;
        rewind(m_NIX);
        fseek(m_NIX, size, SEEK_CUR);
        //jpeg2000 codestream write
        while(size2 > 1) {
                fread(&ch, 1,1, m_NIX);
                fwrite(&ch, 1,1, m_JPC);
                size2--;
        }
        fclose(nix_metadata);    fclose(m_JPC);    fclose(m_NIX);
        ...
}
```

Fig. 5. A part of codes written for NIX file importing module.

cryptographic modes, and it has ODBC and JDBC drivers, so it can be accessed from a variety of clients, even proprietary, fulfilling all the requirements of OGC specifications (Matthew and Stone, 2005; Ramsey, 2006; Paolo *et al.*, 2006; Paul, 2007; Han and Lee, 2009).

PostGIS is a geo-spatial extension to PostgreSQL which allows very powerful geospatial analysis and storage possibilities. According to the recent OTB discussion paper (http://wiki.orfeo-toolbox.org/index.php/OTB-PostGIS_Interface), the ability for OTB to communicate to data base is of major interesting point for the future extensions. Among several applicable data base, PostGIS is the first target, because it seems to be the more widely accepted among the open source alternatives and a generic implementation of the interface between OTB and PostGIS should make possible to switch data bases without major problem.

Fig. 6 shows user interface for Image to database registration in OTB application module. But this function is limited to vector data registration. As of the early 2010, this connection to PostGIS is on developing stage, according to the simple experiment. OTB suggests that this module is useful for building extraction with Label_Object_Map and thresholding, road database updating followed by road detection in OTB, or counting trees along the roads.

In this work, connecting module is newly implemented, regardless of OTB-PostGIS_Interface in OTB. For an example case of actual processing, a kind of feature extraction in OTB is first performed for forest region detection using region growing algorithm. The region growing algorithm is an effective approach for image segmentation. The pixels of seed region, neighbouring a given region are evaluated to determine if they should be considered part of the object. If satisfied, they are added to the region and the process continues as long as new pixels are added to the region, depending on the

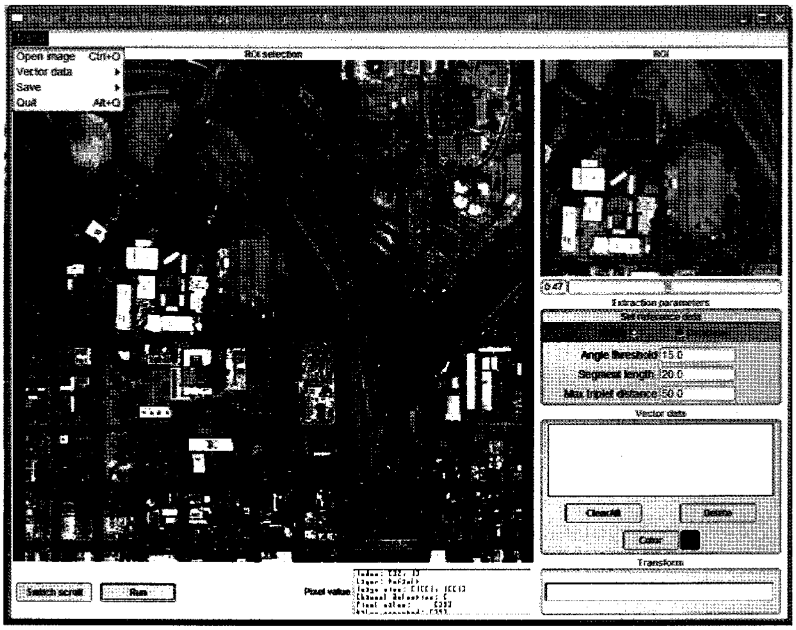

Fig. 6. User interface for Image to database registration in OTB application module.

criteria used to decide whether a pixel should be included in the region or not. In OTB, region growing segmentation algorithm is performed using several methods (OTB Development Team, 2010): Connected_Threshold_Image_Filtering, Curvature_Flow_Image_Filtering, Set_Seed, and Add_Seed.

Fig. 7 is an executing case of forest segmentation by region growing on an airborne image of NIX format, and Fig. 8(A) is another case of urban structure extraction. Fig. 8(B), (C) and (D) represent implemented results for connection to PostGIS; embedded menu in OTB-application, connecting window to PostGIS, and data storage and query

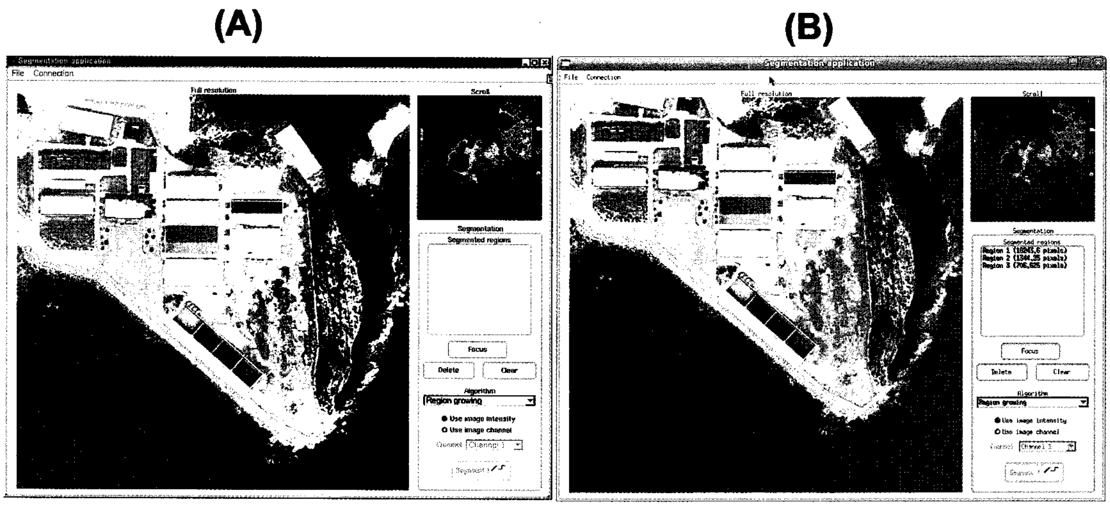

Fig. 7. An example of OTB applications: Segmentation of forest part by region growing algorithm in OTB.

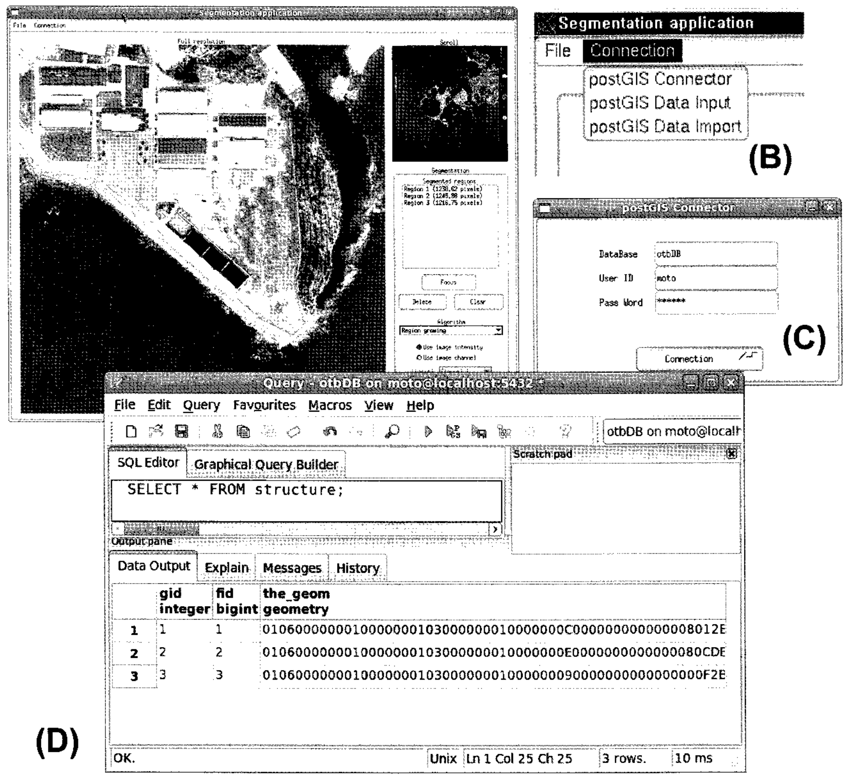

Fig. 8. A part of urban structure extracted by region growing segmentation, and database connection of extracted features: (A) Feature extraction, (B) Connecting menu in OTB, (C) Connector to PostGIS, and (D) Database query of registered features.

```
//save Vector Data : otb -> database.
std::string SegmentationApplicationModel::RetrieveSegmentationResultDataBase(PGconn *conn,
std::string filename)
{
    PolygonListPointerType newPolygonList = m_VectorizationModel->GetPolygonList();
    PolygonListType::Iterator pit = newPolygonList->Begin();
    ...
    ...
    PolygonPointerType newPolygon = PolygonType::New();
    for (PolygonType::VertexListConstIteratorType vit = pit.Get()->GetVertexList()->Begin();
    vit!=pit.Get()->GetVertexList()->End();++vit)
    {
        //============ database MULTIPOLYGON input style ============
        //INSERT INTO Table_Name VALUES ( .... GeometryFromText
        //('MULTIPOLYGON(((1.1 2.2, 3.3 4.4, 5.5 6.6, 1.1 2.2)))', -1));
        // GeometryFromText : string -> binary
        //=========================================================
        PolygonType::VertexType newVertex = vit.Value();
        newVertex[0]+=m_Region.GetIndex()[0];
        newVertex[1]+=m_Region.GetIndex()[1];
        newPolygon->AddVertex(newVertex);
        ...
        ...
    }
    ++pit;
    location = location + xLocation0 + " " + yLocation0;
    //location example
    //5618 8262.5, 5649.5 8289.5, ........ , 5618 8262.5
    //database insert query.
    char query_string[1000];
    sprintf(query_string,"INSERT INTO %s VALUES (%d,%d,GeometryFromText('MULTIPOLYGON(((%s)))',-
    1));",filename.c_str(),rows_number,rows_number,location.c_str());
    res = PQexec(conn, query_string);
    ...
}
```

Fig. 9. A part of codes for database storage of feature extracted from OTB.

processing with respect to extracted features in PostGIS, respectively.

Fig. 9 is a part of code for database storage of feature extracted from OTB, and other additional modules for database connecting and for importing to OTB from database also are implemented.

4. Concluding Remarks

In this study, three kinds of OSRS software satisfied with certain conditions are considered: OSSIM, Opticks, and OTB. After tentative comparative works, OTB is chosen as the most applicable open source. For general users, OTB has beneficiary points for remote sensing application development, due to dual access by full function modules and integrated interfaces. Furthermore, it also useful engine software for developer handling remote sensing data sets. Two implementation of this work are module for importing NIX format, Korean National Geo-based image exchange format based on JPEG 2000 and database, PostGIS/PostgreSQL, connecting module running on OTB. As for an example case, NIX formatted airborne image is used to segmentation process in OTB, and then the results are stored and retrieved in PostGIS database toward more practical applications and ease of uses.

Nowadays, needs for specialized function and more complex application dealing with geo-based images are on the increasing period and some OSRS software meets these trends. Therefore, beyond power users' side of proprietary software, local customization and algorithm development using OSRS software are crucial to build on-demand applications from the viewpoint of developers.

Acknowledgements

This research was supported by a grant by Satellite Information Application Supporting Program of Satellite Information Research Institute in Korean Aerospace Research Institute in 2010

References

Adams, M. D., 2006. *JasPer Software Reference Manual*, Version 1.900.0.

Benthall, S., 2009. An Open Source Web GIS Solution-The OpenGeo Stack, *GEO informatics*, 12: 40-43.

Christophe, E., J. Inglada and A. Giros, 2008. Orfeo Toolbox: A Complete Solution for Mapping from High Resolution Satellite Images, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII. Part B4.

Christophe, E. and J. Inglada, 2009. Open source remote sensing: increasing the usability of cutting-edge algorithms, *IEEE GRS Newsletter*, March: 9-15.

Daoyi C., S. Shams, C. Carmona-Moreno, and A. Leone, 2010. Assessment of Open Source GIS Software for Water Resources Management in Developing Countries, *Journal of Hydro-environment Research*: 1-12, in press.

Hall, G. B. and M. G. Leahy, 2008. *Open Source Approaches in Spatial Data Handling*, Springer.

Han, S.-M. and K. Lee, 2009. Implementation of User Interface for Geo-spatial Information Processing Toolkit using Open Source-based PostGIS, *Korean Journal of Remote Sensing*, 25(2): 185-292.

Inglada, J., E. Christophe, S. Marzocchi-Polizzi, H. De Boissezon, and B. Boissin, 2008. ORFEO *Toolbox: An Open Source Library of Image Processing Algorithms for SAR and Optical High Resolution Images*, Materials at ESA-EUSC 2008 Meeting.

Inglada, J. and G. Merciery, 2008. *Change Detection with Misregistration Errors and Heterogeneous Data through the Orfeo Toolbox*, Technical Report of TELECOM Bretagne # RR-2008003-ITI.

Guzzonato, E., C. Valladeau, and J. Inglada, 2009. Orfeo ToolBox: An Open Source Tool for High Resolution Image Processing, *GeoCap Meeting*.

Markus N., 2009. From a niche to a global user community: Open Source GIS and OSGeo Fondazione E Mach - CRI, *Proceedings of Opensource Geospatial Research Symposium*.

Matthew, N. and R. Stones, 2005. *Beginning Databases with PostgreSQL from Novice to Professional*, Apress.

OTB Development Team, 2010. *The ORFEO Tool Box Software Guide, Updated for OTB-3.2*, 676p.

Özdemir B., S. Aksoy, S. Eckert, M. Pesaresi, D. Ehrlic, 2010. Performance measures for object detection evaluation, *Pattern Recognition Letters*, 31: 1128-1137.

Paolo C., E. Venturato and M. Napolitano, 2006. Free, Cheap, and Powerful PostGIS, the Open Source geodatabase, *GIS Development Middle East*, 2(5): 34-39

Park, Y. J. and K. Lee, 2008. Mashup Application for Geo-spatial Feature Generation on Web Browser using Google Maps API, *Korean Journal of Remote Sensing*, 24: 389-396.

Park, Y. J. and K. Lee, 2009. Application of Deegree of Open Source Middleware to Geo-Portal Implementation, *Korean Journal of Remote Sensing*, 25(4): 367-374.

Paul R., 2007. Introduction to PostGIS, *presented at the FOSS4G2007 Workshops*.

Purvis, M. J., Sambells, and C. Turner, 2006. *Beginning Google Maps Applications with PHP and Ajax*, APRESS.

Ramsey, P., 2006. *PostGIS Case Studies*, Refraction Research.

Rosario R., 2009. OrfeoToolbox: open source information extraction tools for high resolution images, *Proceedings of International Opensource Geospatial Research Symposium*.

Simler, C., C. Beumier, C. Leignel, O. Debeir, and E. Wolff, 2009. The use of Orfeo Toolbox in the context of map updating, *Proceedings of IGARSS 2009*.

Steiniger, S. and E. Bocher, 2009. An overview on current free and open source desktop GIS developments, *International Journal of Geographical Information Science*: 1345-1370.

Steiniger, S. and G. J. Hay, 2009. Free and Open Source Geographic Information Tools for Landscape Ecology, *Ecological Informatics*, 4: 183-195.