# Use of Open Source Programs to Create a Foundation for Developing Serious GIS Application on Mobile Device

**Geoffrey Y.K. SHEA and Jiannong CAO, Hong Kong SAR, China**

**Key words**: mobile GIS, mobile computing, mobile geospatial database, open source program

## SUMMARY

The increasing need of using maps to communicate and display geospatial information for location-based services and mobile applications has brought many challenging issues to developers. Constrained with slow device computing power, less local storage space, lack of efficient data representation structure, developers have to find an appropriate solution in order to develop a mobile geospatial application housed with a rich set of geospatial analysis functions. Although some methods have been used to relax the constraints to certain extend, e.g. representing spatial data by static images with different pre-set scales, they are mostly vendor proprietary and not adaptive to different application for mobile computing in heterogeneous networking environment.

A package of software (set of dynamic link libraries) comprising of 6 open source programs targeting mobile device with Microsoft Windows Mobile operating system was complied successfully during the system prototyping process. This package is the first of its kind in mobile GIS community that made available to public via Google Groups. This set of DLLs formed the foundation for serious mobile GIS development on mobile device. The geospatial data storage and retrieval operations that handled by SpatiaLite are following the specification of the Open Geospatial Consortium (OGC) that recommends a set of SQL environment for GIS operations.

The successful testing of a series of complex spatial analysis query on mobile device using the pre-complied DLLs concluded that the proposed mobile GIS engine is viable for designing and developing a full functional mobile GIS with good performance in handling spatial data query, doing complex spatial analysis operation, and providing satisfactory user-interaction.

# Use of Open Source Programs to Create a Foundation for Developing Serious GIS Application on Mobile Device

**Geoffrey Y.K. SHEA and Jiannong CAO, Hong Kong SAR, China**

## 1. CURRENT DEVELOPMENT AND RELATED WORKS IN MOBILE GIS

Mobile GIS applications are designed to communicate geospatial information between human user and servers (both application server and database server) via mobile device either in connected mode or disconnected mode. The information to be exchanged depends on the type of the application being developed. While some applications require the transmission of maps corresponding to the current position of the mobile user, others require the transmission of non-spatial information. Basically the works carried out by researchers in mobile GIS can be categorized into four major areas: (1) Disconnected Mobile GIS applications; (2) Mobile spatial database modeling; (3) Mobile location services; and (4) Mobile positioning/navigation. Communication is successful if the receiver interprets the information in the manner intended by the sender (Shannon 1949).

Because of the advances in communications, positioning, and mobile computing technologies, now the opportunities for delivering maps over the Internet s not limited to wired connection, but further extending Internet GIS to incorporate geo-referenced information into valuable user-friendly solutions for the average mobile user – what we called "mobile location services". Most of these mobile location services involve communication with geospatial database server via diversified wireless network, such as Wi-Fi, GPRS, SMS/MMS, 3G, etc. In order to differentiate from the Internet GIS, the term mobile GIS has developed to mean solutions dedicated to mobile devices that integrating various technology components drawn from positioning, GIS, mobile computing and communication.

It is not a surprise to find more than 30 different web mapping software available in the market. They all use their own data format for storing information and their own query syntax for retrieving data. Most of them are mainly developed for the desktop environment and only a small number of them are designed for mobile devices. Because of the differences in data format, it is difficult to share data directly between disparate systems. Therefore, it is one of the challenges high in the list to develop an open and extendable geospatial data model specifically for mobile environment that can offer geospatial data readily used and shared by a variety of mobile applications.

In order to develop open, extensible and adaptive mobile GIS system using open source software, it is important to understand the basic principles of the underlying technologies first. The next step of this paper is to further illustrate the feasibility of setting up a full functioning mobile GIS with the underlying concepts and technology from a technical and practical point of view.

## 2. OpenGIS SIMPLE FEATURES SPECIFICATIONS FOR SQL

The document "OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 2: SQL option" (OGC 2006) published by the Open Geospatial Consortium defines a number of conceptual ways for extending an SQL RDBMS to support geometry types for spatial data creation, storage, manipulation in a standard SQL environment and in an OpenGIS conformant manner.

### 2.1 SQL Geometry Type hierarchy

The set of geometry types proposed by OGC is based on the OpenGIS Geometry Model. Each geometry type in this model will bear two general properties: (1) the spatial referencing information; and (2) spatial representation in Euclidean plane.

The root type of this geometry type hierarchy is Geometry. It has four subtypes for Point, Curve, Surface and Geometry Collection. LineString is a subtype of Curve and Polygon is a subtype of Surface. MultiPoint, MultiLineString and MultiPolygon are subtypes of Geometry Collection. All the types are classified either as instantiable or non-instantiable. Geometry, Curve and Surface are non-instantiable types.

### 2.2 Supported spatial data representation formats: Well-Known Text and Well-Known Binary

The geometry feature can be represented either in Well-Known Text (WKT) format or Well-Known Binary (WKB) format.

The WKT format is used to exchange geometry feature in ASCII form, such as POINT(802345.0 835678.9). A Backus-Naur grammar that defines the formal rules for producing WKT values can be referred to the above-mentioned OGC specification.

The WKB format is used to exchange geometry feature in binary streams that represented by BLOB value in a geometry column. Using WKB format in all spatial computation definitely will gain better performance over WKT.

### 2.3 Spatial metadata

In the OpenGIS specifications, a schema is defined for the management of spatial metadata in an SQL environment that support geometry features. Basically, the spatial metadata are implemented as two database tables: (1) SPATIAL_REF_SYS – table describes the coordinate system and transformations for geometry; and (2) GEOMETRY_COLUMNS – table stores a list of geometry columns as defined in the database. The purpose of this spatial metadata is to ensure the spatial DBMS is reliable to provide consistent geometry data by enforcing some appropriate constraints on the spatial data. Result of spatial search or operation should be presented in a well defined coordinate environment via the use of some standard spatial reference systems, such as European Petroleum Survey Group (EPSG).

## 3. OPEN SOURCE DBMS SUPPORTING GEOSPATIAL DATA

There are numerous choices of relational database management system (RDBMS) in the open source community to serve the typical record query. However, the choice is limited if the database engine possesses the capability to store, handle and manipulate geometry features with spatial referencing information. At the time of this writing, all of the DBMSs available in the open source community that are empowered to handle geospatial data are DBMS that offer an SQL environment that has been extended with a geometry-valued SQL column and a number of spatial referencing system tables. Normally, the DBMS will only provide a small number of spatial data handling functions to facilitate itself to store and retrieve spatial data correctly and properly. Other data query or analysis functions on geometry types will be provided by other extensions, such as PostGIS, SpatiaLite. There are three widely used open source DBMSs supporting geospatial data: (1) PostgreSQL + PostGIS; (2) MySQL Spatial Extension; and (3) SQLite + SpatiaLite.

## 4. BUILDING A FULL FUNCTION MOBILE GIS SOFTWARE FOUNDATION USING OPEN SOURCE

It is revealed that the development of a min-sized GIS equipped with a full set of spatial query and analysis function for mobile device is far behind the development for desktop environment and lack of a software development package for mobile GIS platform which is publicly available in only source community.

It is not difficult to spot a bunch of software tools in the open source projects for developing desktop GIS software such as PostgreSQL and PostGIS. Therefore, this section is trying to explore a set of open source projects which are suitable for transforming to resource limited mobile device as the software foundation for developing a full function mobile GIS application.

As mentioned in Section 3 above, there are three widely used open source DBMSs supporting geospatial data for critical consideration. After some tests and comparisons, the SQLite + SpatiaLite combination is selected to go for further evaluation on mobile device platform. The selection of such combination is based on three reasons: (1) Small software footprint; (2) File-based RDBMS; and (3) Support OGC specification on Simple Feature for SQL.

This mobile GIS software foundation package, we name it libMobileGIS collectively, provides a set of pre-compiled dynamic link libraries (DLLs) specifically for Microsoft Windows Mobile-based device. The libMobileGIS is aimed for developing serious GIS applications on mobile device. In the package the geospatial data storage and retrieval are handled by SQLite and SpatiaLite together. Core GIS manipulation and analysis functions are provided by PROJ.4 and GEOS. Text encoding is handled by libiconv. The geospatial data storage and retrieval operations that handled by SpatiaLite are following the specification of the OGC, which recommends a set of SQL environment for GIS operations.

### 4.1 libMobileGIS components

Altogether there are six open source projects included to form the mobile GIS software foundation, namely SpatiaLite, SQLite, PROJ.4, GEOS, libiconv, and WCELIBCEX.

### 4.2 Output of compilation and build process

The compilation and build processes are using Microsoft Visual Studio 2008 Professional edition. The target device is Windows Mobile 5 and 6 with Microsoft .NET Compact Framework 3.5. After the compilation and build processes, six binary files will be produced: (1) WMcharset-1.2.dll; (2) WMgeos_c-3.1.0.dll; (3) WMiconv-1.9.2.dll; (4) WMproj-4.6.1.dll; (5) WMspatialite-2.3.0.dll; and (6) WMsqlite-3.6.13.dll. All the DLLs should reside in the same directory on the smart device in order to develop functioning application with these pre-compiled libraries.

The binary files are available at http://myweb.polyu.edu.hk/~lsykshea/public/spatialite-related/.

### 4.3 Problems encountered

The original sources of libMobileGIS are coming from open source projects and they all targeted for Linux and Microsoft Windows operating systems. There is no difficulty in the compilation process when performed in Linux and Windows. However, the story is totally different for the compilation on Windows Mobile platform. As we know mobile device is equipped with less powerful CPU, less memory space, and limited storage. Therefore, the transformation from the desktop to mobile device is frustrating (at least in the beginning) and the compilation of those C- and Java-base source programs is another challenging task for a Visual Basic programmer. A lot of problems have been encountered during the compilation, modification and debugging phases before coming to the success compilation.

Since Windows Mobile was built to be small as the first design goal of the Windows Mobile development team. That means there are a number of libraries and functions intentionally missed out from Windows Mobile. Therefore, the first difficult hurdle to overcome is to fix or find an alternative for the dependency of a missing header file required in the original source. Basically, WCELIBCEX project serves this purpose by providing most of the missing header files for use in Windows Mobile.

As the libMobileGIS is the integration of 6 independent open source projects. The modification of one internal configuration settings might affect the others. How to get a balanced and workable configuration setting on all the involved projects and work out a proper project dependency order is the second hurdle to be solved. After numerous tests on this issue, the proper and success build order for libMobileGIS is found to be in this sequence 1) WCELIBCEX, 2) libiconv, 3) PROJ.4, 4) GEOS, 5) SQLite, and 6) SpatiaLite.

### 4.4 Results achieved

A full function mobile GIS means there is a software package providing the necessary development environment for developers to work on the geospatial data storage, handling, manipulation and communication. The successful compilation of libMobileGIS in this experimental test provides a new stage for such development which has been long awaited in the mobile GIS community. The release of libMobileGIS to public would set an important milestone in the mobile GIS community and through libMobileGIS a rich set of spatial operations and analysis functions are exposed to mobile GIS developers.

## 5. GENERAL GEOSPATIAL OPERATIONS TEST

The libMobileGIS developed in Section 4 above lay the foundation for building a full function mobile GIS application. In order to test the capability and extend of functionality exposed by libMobileGIS, a test suite containing three parts (namely Test Part A, Test Part B, and Test Part C) is designed to be conducted on a Windows Mobile-based device.

### 5.1 Common geospatial operation capability test

The first part of the test suite is focused on those common geospatial operations supported by libMobileGIS. Altogether 24 spatial operations have been tested in this part: 1) IsValid, 2) IsSimple, 3) IsRing, 4) Area, 5) Distance, 6) GLength, 7) Centroid, 8) PointOnSurface, 9) Boundary, 10) Buffer, 11) ConvexHull, 12) Equals, 13) Within, 14) Contains, 15) Disjoint, 16) Touches, 17) Overlaps, 18) Crosses, 19) Intersects, 20) GUnion, 21) Intersection, 22) SymDifference, 23) Difference, and 24) Simplify.

### 5.2 Common geospatial analysis functions test

The second part of the test suite is focused on the following objectives: 1) Formulate valid spatial query in standard SQL sytax; and 2) Determine the extent of geospatial analysis function supported.

### 5.3 Advanced geospatial analysis functions test

The last part of the test suite is conducted to find out the performance (expressed in terms of elapse time) of the following complex geospatial analysis functions: 1) Point-in-polygon test; 2) Polygon-in-polygon test; 3) Point/polygon overlay; and 4) Polygon/polygon overlay.

The SQL statements employed to get the elapse time for test part C are shown in the following table.

| | |
|---|---|
| 1. | Count the total records |
| | `SELECT count(*) FROM my_bg_spatial;`<br>`SELECT count(*) FROM my_sg_spatial;`<br>`SELECT count(*) FROM my_rg_spatial;` |
| 2. | Number of BLDG within polygon (without MBR filter) |
| | `SELECT count(*)`<br>`FROM my_sg_spatial t3, my_bg_spatial t4`<br>`WHERE (t3.site_code = 78382)`<br>`AND (contains(t3.site_geom, t4.bldg_geom));` |
| 3. | Number of BLDG within MBR |
| | `SELECT count(*)`<br>`FROM my_sg_spatial t1, my_bg_spatial t2`<br>`WHERE t1.site_code = 78382`<br>`AND MbrContains(t1.site_geom, t2.bldg_geom);` |
| 4. | Number of BLDG within polygon (use MBR filter) |
| | `SELECT count(*)`<br>`FROM my_sg_spatial t3, my_bg_spatial t4`<br>`WHERE (t3.site_code = 78382)`<br>`AND (t4.pk_uid IN (`<br>`   SELECT t2.pk_uid`<br>`   FROM my_sg_spatial t1, my_bg_spatial t2`<br>`   WHERE (t1.site_code = 78382)`<br>`   AND (MbrContains(t1.site_geom, t2.bldg_geom))))`<br>`   AND (contains(t3.site_geom, t4.bldg_geom));` |
| 5. | Polygon-In-MBR Test |
| | `SELECT t1.cacode`<br>`FROM hk01.constituencies t1, my_sg_spatial t2`<br>`WHERE (t2.site_code = 78382)`<br>`AND (mbrwithin(t2.site_geom, t1.geom));` |
| 6. | Polygon-In-Polygon Test |
| | `SELECT t1.cacode`<br>`FROM hk01.constituencies t1, my_sg_spatial t2`<br>`WHERE (t2.site_code = 78382)`<br>`AND (Within(t2.site_geom, t1.geom));` |
| 7. | Polygon-In-Polygon Test |
| | `SELECT Count(*)`<br>`FROM constituencies t1, my_sg_spatial t2`<br>`WHERE (t1.cacode = 'M26')`<br>`AND (t2.site_code BETWEEN 70000 AND 79999)`<br>`AND (Contains(t1.geom, t2.site_geom));` |

### 5.4 Results achieved

All the three parts of the test suite are conducted successfully on a Windows Mobile-based device with hardware configuration as shown in Figures 1 to 3 in Appendix. The successful completion of the test suite demonstrated that libMobileGIS is functioning as design on providing functionalities both for basic and advanced operations in mobile environment. There is no difficulties using standard RDBMS (here refers to SQLite) in handling large volume of records in a standard table irrespective of geometry type such as point, line string or polygon. The screen shots of selected results for the test suite are shown in Figures 5 to 16 in Appendix.

#### 5.4.1 Analysis of test part A results

The Test Part A results revealed that the libMobileGIS is functioning as design on the basic GIS operations on Windows Mobile-based device. The positive results achieved in this part put forward the other rigorous and complex geospatial analysis tests on mobile device.

#### 5.4.2 Analysis of test part B results

The test results achieved in Test Part B illustrated that it is not a problem to retrieve attributes of geospatial data from a standard RDBMS via different means of interaction, for example, clicking on an item from a table user interface or tapping anywhere on the screen. That means it is the job of application developer to design an appropriate user interface to interact with the database repository via the functions provided by libMobileGIS.

#### 5.4.3 Analysis of test part C results

The seven testing scenarios designed for part C are the most difficult GIS analysis functions to be fulfilled even within desktop PC environment. It is a good and positive sign to receive a successful completion of test on this part. The results reflect that even a mobile device equipped with less computing power and memory size can perform a set of reasonably good GIS analysis functions comparable with desktop PC environment.

The results showed in Figure 14 in Appendix indicated that the use of a geospatial feature filter, minimum bounding rectangle technique, would substantially improve the performance of spatial query by nearly a factor of ten in a mobile environment. This also indicated that a geospatial feature filtering technique, such as minimum bounding rectangle, should be employed as far as possible for all complex analysis functions in order to achieve a good computation performance.

## 6. DEVELOPMENT OF A GENERIC GEOSPATIAL DATA EDITOR

A variety of DBMS are available for mobile devices, but none of them support geospatial data. Thus, there is a need to develop a generic geospatial data editor targeted for mobile devices to fill-up the gap. This generic geospatial data editor has two primary tasks to consider in program design stage: (1) providing accurate display of geometry feature in descriptive and graphic output; and (2) providing data manipulating functionalities match up with its desktop counterpart.

### 6.1 amGIS.SQLite

amGIS.SQLite is the first mobile GIS application based on libMobileGIS and it is a program aimed to serve as generic geospatial data editor on mobile devices installed with .NET Compact Framework 3.5. What amGIS.SQLite does provide is support for viewing table definition, all kind of geometry record in descriptive and graphic output, and image blob. It has the ability to navigate record-by-record, page-by-page, or go to the beginning/end of database.

What functionalities are missing from amGIS.SQLite at this stage of development are: no data editing function and no support of encoding language other than English.

### 6.2 Results achieved

The most important achievements in this software development is the successful implementation of a power generic geospatial data viewer capable of showing data accurately irrespective of data size and data type, such as a large database size of 107 megabytes and a polygon record comprising of 125007 vertices (refers to Figures 17 and 18 in Appendix for the screen shot showing the details of the polygon with record id = 1).

The time for showing the details of such a complex geometry feature is just a matter of 2 to 3 seconds which is rated highly satisfactory. At the time of this writing, even the most popular free SQL Server CE Edition can do nothing when feeding with geospatial data.

Because SQLite databases are files in the Windows CE file system, they can be accessed programmatically using the File and Directory classes provided by .NET Compact Framework. As such, we can easily copy a database file, delete a database file, or determine the existence of a database file programmatically in a similar fashion as other file and directory access. If a simple database file protection is desired, an RC4 encryption algorithm can be applied onto the database file.

Generally, this software demonstrated the capabilities in 1) handling geospatial database with ease, 2) providing accurate data both in textual and geometry, 3) providing database viewer functionality comparable with its other common desktop counterparts.

## 7. CONCLUSION AND RECOMMENDATION

A number of experimental tests, geospatial data preparation, and evaluations have been conducted. A full function mobile GIS software foundation, libMobileGIS, using open source is produced for developing mobile GIS applications. Three geospatial databases with varying file size have been created for subsequent geospatial analytical tests on mobile devices using libMobileGIS libraries.

The successful completion of a set of three geospatial analytical tests demonstrated that libMobileGIS is functioning as design and revealed that complex geospatial analysis functions such as polygon overlay is not only found on desktop PCs but also available on mobile devices.

## 8. ACKNOWLEDGEMENTS

Acknowledgement is due to The Hong Kong Polytechnic University for the financial assistance of this study. Thanks also go to the Lands Department of The Government of the Hong Kong Special Administrative Region (HKSAR) for providing the 1:10000 and 1:20000 of Hong Kong digital map data in Arc/Info export format (.e00) and geo-reference data (BG1000, RG1000 and SG1000) in text format for this project.

## REFERENCES

OGC, 2006. OpenGIS Implementation Specification for Geographic information – Simple feature access – Part 2: SQL option.

Shannon, C.E., Weaver, W., 1949. The Mathematical Theory of Communication. The University of Illinois Press, Urbana, Illinois.

## APPENDIX

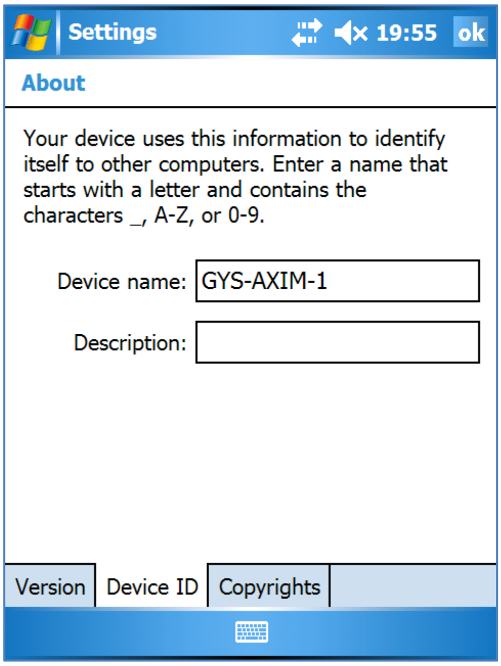


Figure 1 – Mobile device name

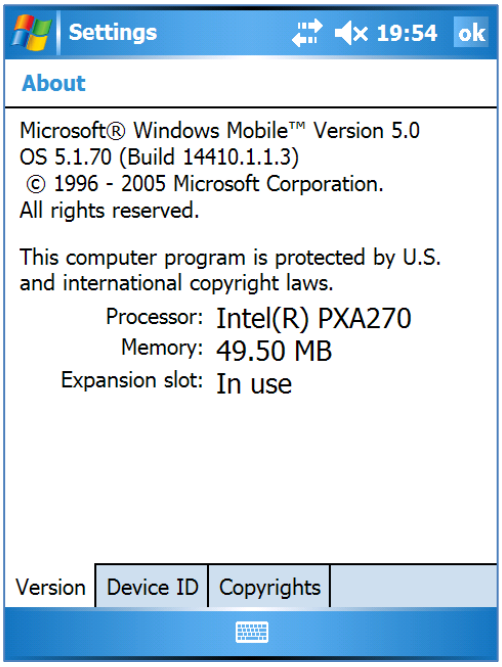


Figure 2 – CPU type and memory size

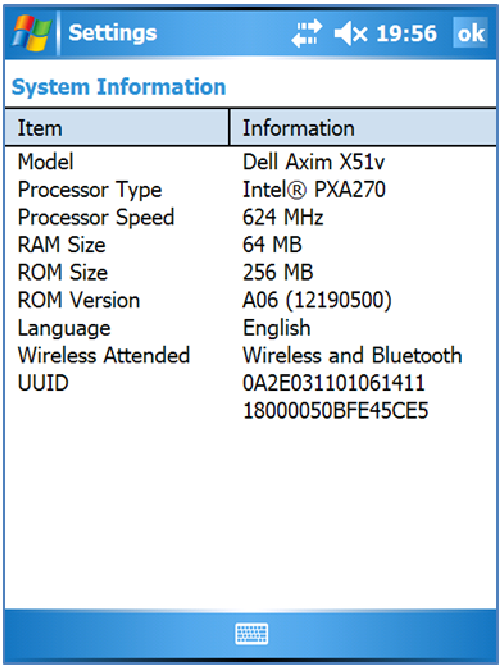


Figure 3 – Mobile device system information

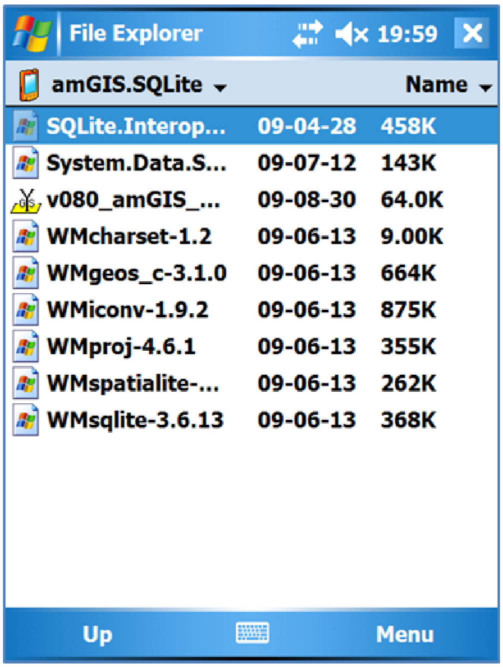


Figure 4 – Show the executable file and its dependencies

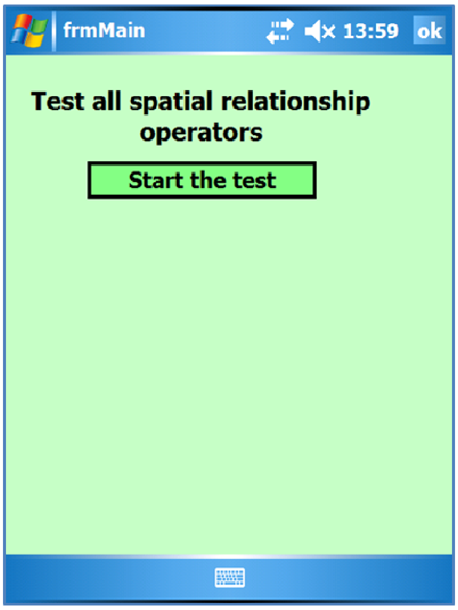


Figure 5 – Click "Start the test" button to run all pre-defined spatial operations

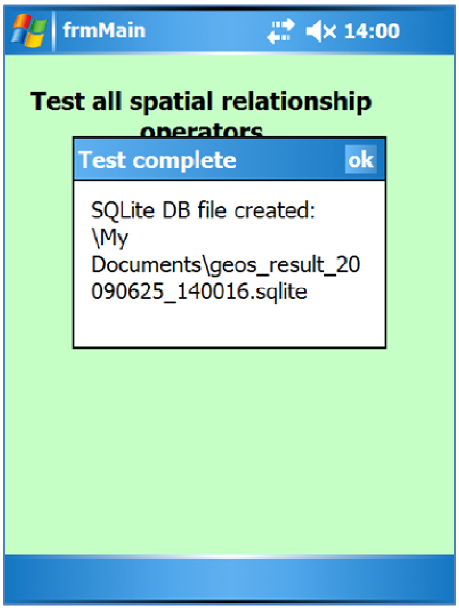


Figure 6 – Test results are stored in a spatial database file as shown

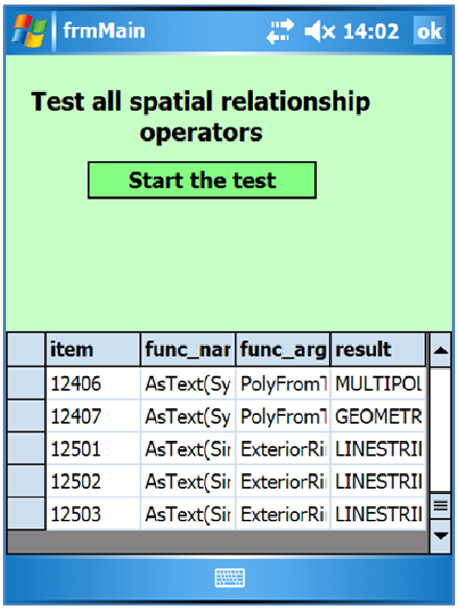


Figure 7 – Results are shown in a datagrid control

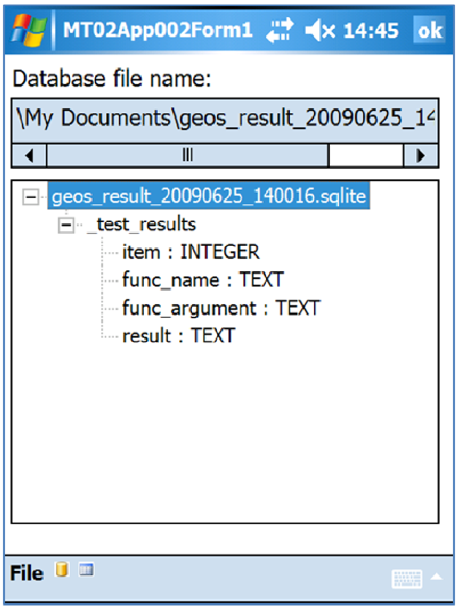


Figure 8 – Database informaton of the newly created DB file

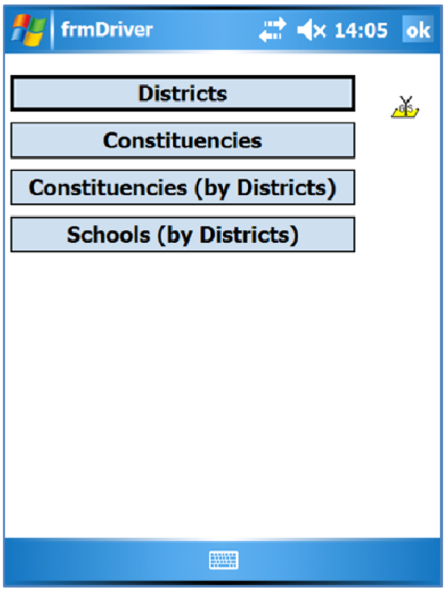


Figure 9 – Test Part B operations selection

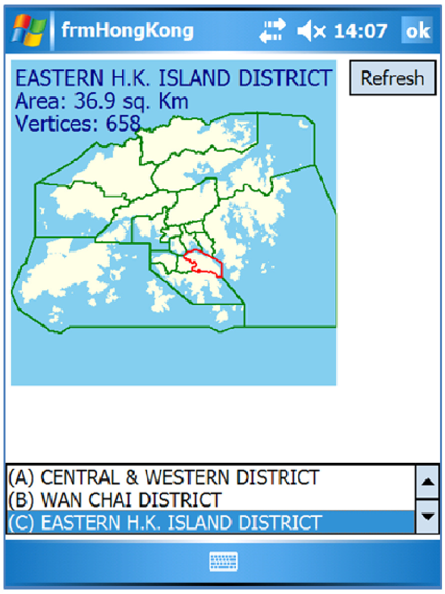


Figure 10 – District boundary is highlighted in red color when tapping anywhere within Eastern HK Island

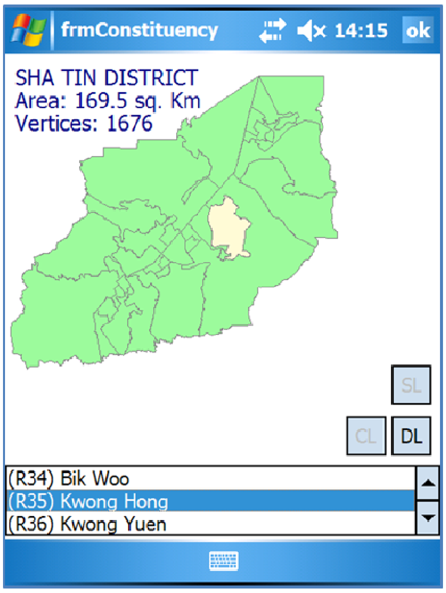


Figure 11 – Constituencies within Sha Tin District are shown

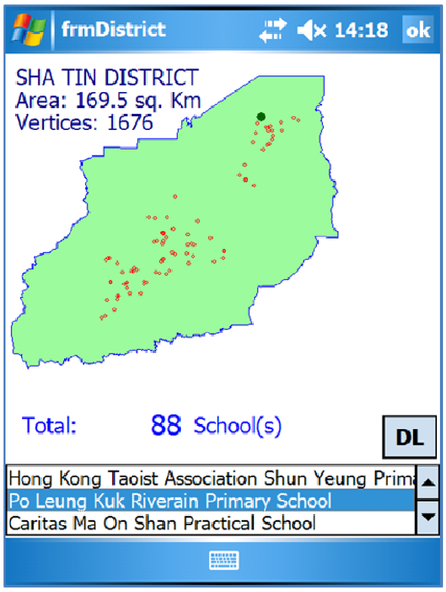


Figure 12 – The selected school is highlighted in green color in the map

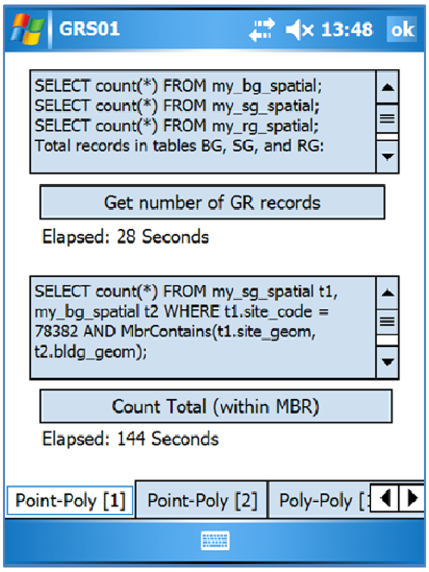


Figure 13 – Time spent to get number of geo-referenced data and point-in-polygon search

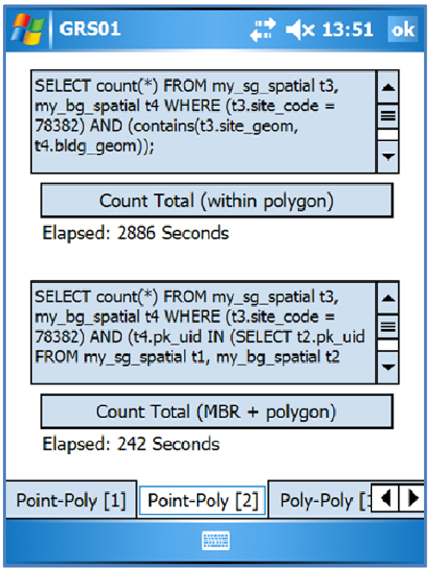


Figure 14 – Time spent to perform another set of point-in-polygon search

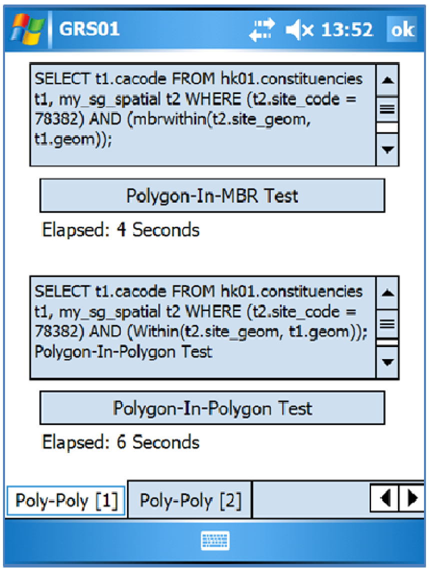


Figure 15 – Time spent to perform polygon-in-polygon search

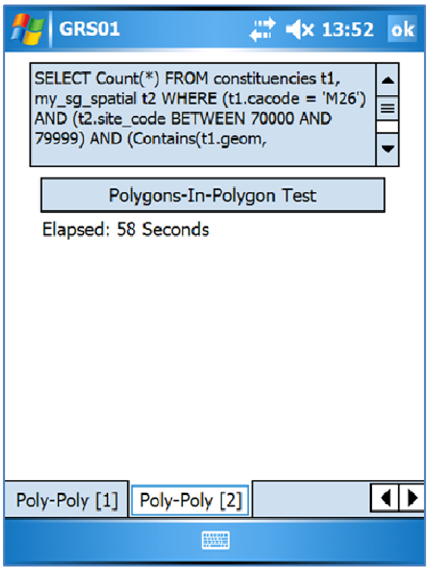


Figure 16 – Time spent to perform another polygon-in-polygon search

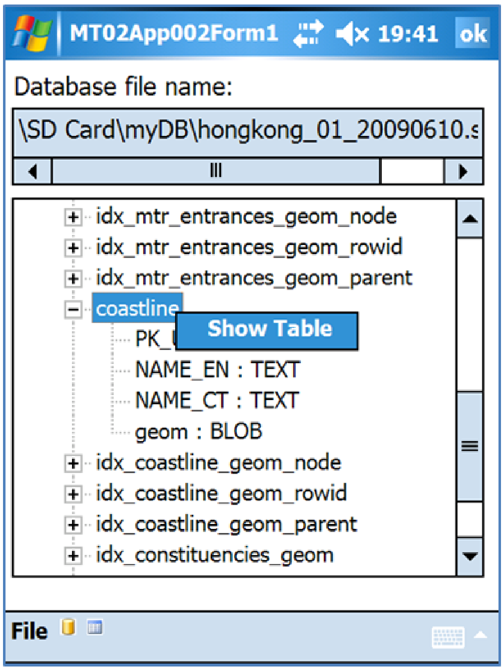


Figure 17 – Select the coastline table

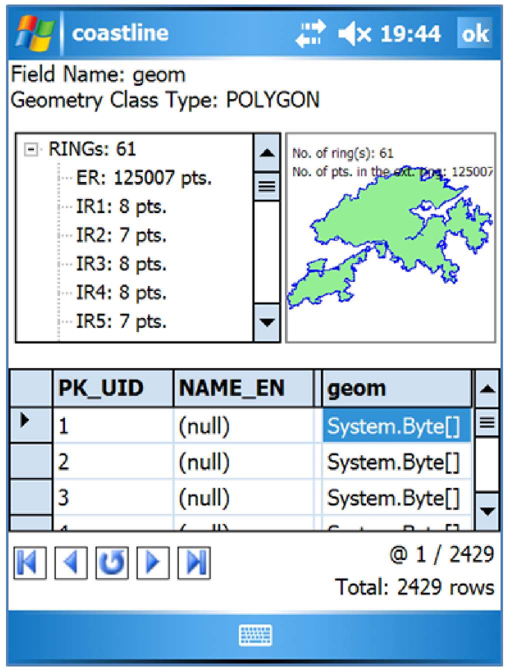


Figure 18 – Attributes of the selected polygon feature are shown (more than 120,000 points)

## CONTACTS


Mr. Geoffrey Y.K. Shea
Department of Land Surveying & Geo-Informatics, The Hong Kong Polytechnic University
Room HJ702, General Office of LSGI, Hung Hom, Kowloon
Hong Kong
Tel. +852 2766 5958
Fax +852 2330 2994
Email: Geoffrey.shea@polyu.edu.hk
Web site: http://www.lsgi.polyu.edu.hk/staff/geoffrey.shea

Prof. Jiannong Cao
Department of Computing, The Hong Kong Polytechnic University
Room PQ806, Mong Man Wai Building, Hung Hom, Kowloon
Hong Kong
Tel. +852 2766 7275
Email: csjcao@comp.polyu.edu.hk
Web site: http://www.comp.polyu.edu.hk/~csjcao