

A SIMPLE AND EFFICIENT SQL-BASED APPROACH FOR RETRIEVAL OF GEOSPATIAL DATA IN MOBILE GIS APPLICATION

G. Y. K. Shea^a, J.N. Cao^b

^a Dept. of Land Surveying & Geo-Informatics, The Hong Kong Polytechnic University, Hong Kong SAR, China – geoffrey.shea@polyu.edu.hk

^b Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China - csjcao@comp.polyu.edu.hk

Commission II

KEY WORDS: Mobile GIS, mobile computing, mobile geospatial database

ABSTRACT:

The core objective of mobile GIS application is to provide vicinity geospatial information to the moving clients based on their current position. Since most of the mobile devices are limited in resources such as processing power, storage capability in terms of memory size, communication speed. Sending lot of map contents are normally required for all those applications as well. Therefore, the success of a mobile GIS application depends mainly on the efficiency of the geospatial data delivery method used and how well the adaptation can be employed. Two customised spatial index numbering schemes have been devised. Evaluation has been conducted in the actual mobile environment to determine the relative efficiency for this two numbering schemes. The main purpose for devising a unique spatial index numbering system is to provide a faster and simpler way other than using R-Tree algorithm to fetch the regularly geo-referenced map base images. A prototype mobile application is developed based on this customised spatial index scheme with an open source spatial-enabled RDBMS engine. The work flows and algorithms in the initial formation and subsequent updating process of the local dynamic moving geospatial database will also be described in this paper.

1. GROUPING OF GEOSPATIAL DATA

Basically, the transmission of map contents to the mobile client is comprised of two kinds of information – geographic and attributes. There is a common set of geospatial data that needs to be served to the mobile clients irrespective of the type of applications. This common set of geospatial data contains: (a) road features; (b) building features; (c) hydrology features such as coastline, rivers and reservoirs; and (d) major point features such as bus stops and address. Usually, we called this set of data as “map base”. Map base data bears the following characteristics:

- Delivered to the client as a single image
- Each single image is geo-referenced (i.e. position information is embedded in the image)

The map base data serves as the backdrop for displaying other geographic related data. Therefore, another set of data in association with the map base will be sending to the client in the same time. We called this set of data as “application data”.

Usually, the application data will contain more descriptive information other than the position information and the number of layers to be included can be one or more.

The format of this set of data is varying and is depending on the type of application. It can be stored in a database, in XML format, in text file format, etc. Figure 1 depicts the geographical relationship between the map base data and application data.

According to George Cantor (1845-1918), the founder of set theory, “A set is a collection of definite, distinguishable objects of perception or thought conceived as a whole”. The map base data can be regarded as a set of object collection containing

geo-referenced images. The notation for this set can be expressed as follows:

$$M = \{m_1, m_2, \dots, m_n\} \tag{1}$$

where m_i is the map base element and $i = 1, 2, \dots, n$

Similarly, the application data, set A, can be regarded as a set of data comprising of two other sets, namely position data, set P, and descriptive data, set D.

$$A = P \cup D \tag{2}$$

Usually, both sets of data M and A will be transferred to the

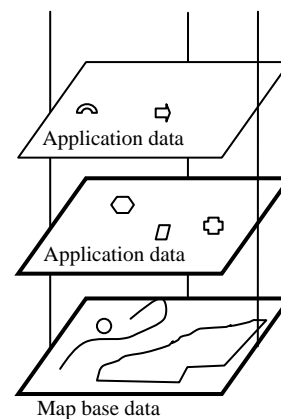


Figure 1. Geographical relationship

client as a whole while the application is executing. That means the client will receive a set of geospatial data, set C, containing both the map base and the application data.

$$C = MUA \tag{3}$$

2. DEFINING A UNIQUE SPATIAL INDEX NUMBERING SYSTEM FOR MOBILE GIS

Two numbering schemes have been devised and tested on their efficiency in the actual mobile environment. The design of unique spatial index number for both schemes is based on a local plane coordinate system as detailed in the following section.

The main purpose for devising a unique spatial index numbering system is to provide a faster and simpler way other than using R-Tree algorithm to fetch the regularly georeferenced map base images. The usual case is to determine the map base tile that contains the input point, which is a spatial search algorithm called point-in-polygon test. Therefore, we are devising an algorithm to find out the spatial index number of the map base tile that contains the input coordinate of a point.

2.1 Hong Kong 1980 Plane Coordinate System

The local plane coordinate system used in Hong Kong is based on Hong Kong 1980 Grid System (HK80GS) that devised by Survey and Mapping Office of Lands Department. The map name convention for scales 1:20000, 1:10000, and 1:5000 used in this system is shown in Figure 2.

The whole of Hong Kong is covered by a series of map with extent measures 60 Km in the x direction and 48 Km in the y direction. The origin of this grid system is located at the lower left corner with coordinate's value (800000, 800000). The x and y axis is named easting and northing respectively.

2.2 Spatial Index Scheme 1

The base image to be partitioned is created from 1:5000 maps, e.g. 13SWA. The ground coverage for this base image is 3750m wide in the x direction and 3000m high in the y direction. The corresponding image size for this base image is 4800 x 3840 pixels. Each base image is partitioned into 15 rows by 15

columns evenly to give a total of 225 small tiles.

Following this partitioning concept, the whole of Hong Kong is covered by 57600 small tiles (16 x 4 x 4 x 225) mesh of 240 rows by 240 columns.

The unique spatial index number for each small tile in this scheme is a 7- or 8-digit integer coming from 5 parts as illustrated in Figure 3.

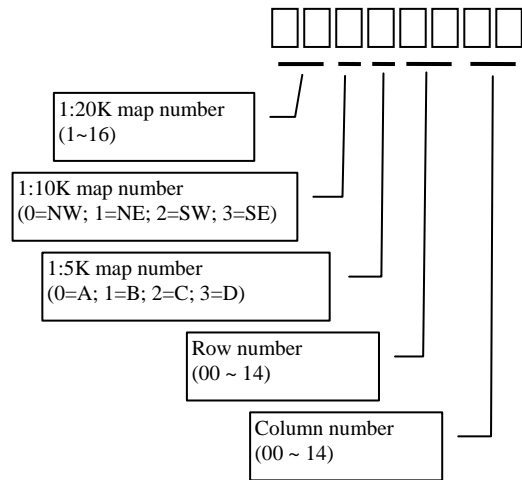


Figure 3. Scheme 1 spatial index number

- The first part, P1, is a 1- or 2-digit number of the corresponding map in scale 1:20000.
- The second part, P2, is a 1-digit number of the corresponding map in scale 1:10000. NW is presented as 0, NE is presented as 1, SW is presented as 2, and SE is presented as 3.
- The third part, P3, is a 1-digit number of the corresponding map in scale 1:5000. Similar as above A is presented as 0, B is presented as 1, C is presented as 2, and D is presented as 3.
- The fourth part, P4, is a 2-digit row number.
- The fifth part, P5, is a 2-digit column number.
- Both row and column number starts at the lower left corner with value 0.

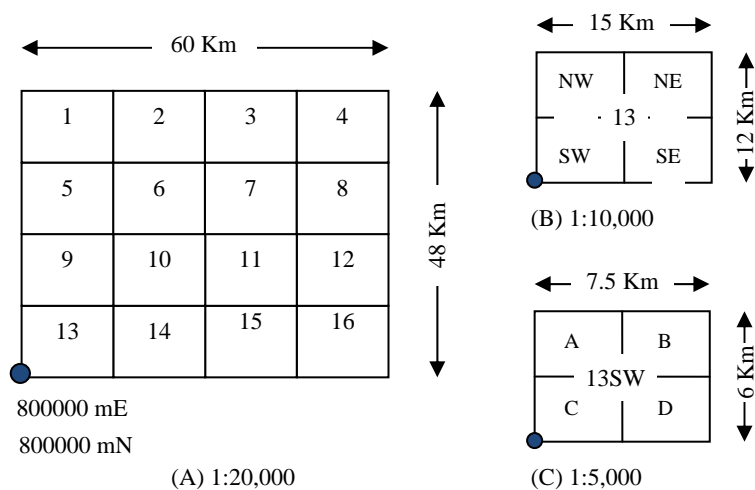


Figure 2. Hong Kong map naming convention

For example, all those small tiles within map 13SWD will be in the range of 13230000 and 13231414 as shown in Figure 4. Similarly, those small tiles within map 13SWA will be in the range of 13200000 and 13201414.

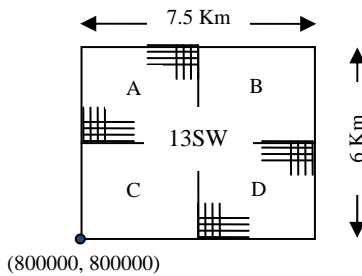


Figure 4. Examples spatial index numbers (Scheme 1)

2.2.1 Finding Scheme 1 Unique Id from Coordinates

The spatial index id is actually depending on a number of factors: (1) point coordinates, (2) relative location in map scale 1:20K, (3) relative location in map scale 1:10K, (4) relative location in map scale 1:5K, (5) relative row number to lower left corner of map scale 1:5K, and (6) relative column number to lower left corner of map scale 1:5K. When express in mathematical function would be:

$$id = f(x, y, P1, P2, P3, P4, P5) \tag{4}$$

For example, a point with coordinates (804550, 802500) will be contained in small image tile with spatial index id equals to 13231203.

2.2.2 Finding Tile Minimum Bounding Rectangle from Scheme 1 Spatial Id

There are two ways to express a minimum bounding rectangle: (1) lower left corner coordinates and upper right corner coordinates, and (2) lower left corner coordinates, width and height of the bounding rectangle. We use the second to store the small image tile minimum bounding rectangle information. The equations to find the lower left corner coordinates are expressed below.

$$X_{tile} = X_{1120K} + X_{1110K} + X_{110K} + 250 \times RowNr \tag{5}$$

$$Y_{tile} = Y_{1120K} + Y_{1110K} + Y_{110K} + 200 \times ColNr \tag{6}$$

The width and height for each of the minimum bounding rectangle is 250m and 200m respectively.

For example, the lower left corner of the MBR for the tile with spatial id equals to 13231203 is (804500, 802400).

2.3 Spatial Index Scheme 2

In Scheme 2, the method of partitioning a large image into a number of small tiles is same as the method used in scheme1 above. That is the whole of Hong Kong will be covered by 57600 small tiles mesh of 240 rows and 240 columns after

partitioning. The difference is the way to number each small tile uniquely.

The unique spatial index number for each small tile in scheme 2 is an 8-digit number coming from the row number and column number of the small tile position in this mesh. The first four digits are designated for row number and the last four digits are designated for column number. The lower left corner of the small tile will have a row number 0 and column 0. The row number is increased upwardly and the column number is increased toward right hand side. In order to avoid leading

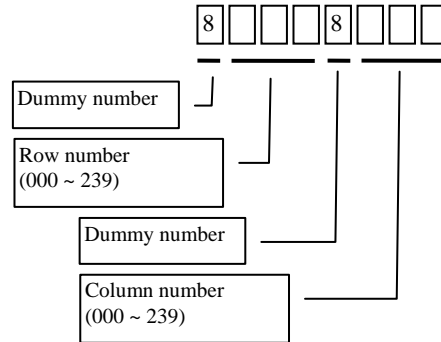


Figure 5. Scheme 2 spatial index number

zeros, both row number and column number are shifted by a value of 8000 (refers to Figure 5).

Therefore, the spatial index number for the lower left corner of the small tile is 80008000, whereas, the upper right corner tile

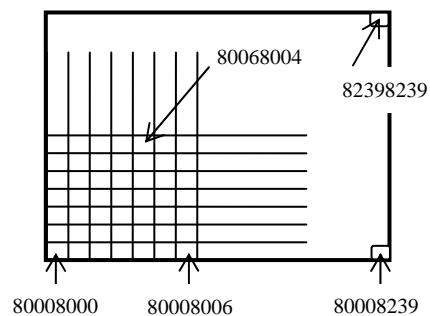


Figure 6. Examples of spatial index number

will have an index number 82398239. Some examples are shown in Figure 6.

2.3.1 Finding Scheme 2 Spatial Id from Coordinates

The spatial id for scheme 2 is comparatively simpler than scheme-1 and is depending on the coordinates of the lower left corner of the small image tile which depends on which row and column band that the small tile falls into. When express in mathematical function would be:

$$RowNr = f(y, \text{multiples of } 200) \tag{7}$$

$$ColNr = f(x, \text{multiple of } 250) \tag{8}$$

Using the same example point in scheme 1, i.e. (804550, 802500), the point is contained within the small image tile with spatial index id equals to 80128018.

2.3.2 Finding Tile Minimum Bounding Rectangle from Scheme 2 Spatial Id

The equations for determining the lower left corner coordinates are expressed below:

$$x_{tile} = 800000 + ColumnNr \times 250 \quad (9)$$

$$y_{tile} = 800000 + RowNr \times 200 \quad (10)$$

The width and height for each of the minimum bounding rectangle is 250m and 200m respectively.

For example, the lower left corner of the MBR for the tile with spatial id equals to 80128018 is (804500, 802400).

2.4 Analysis of the Two Index Schemes

Theoretically, investigating the equations given in Sections 2.2.1 and 2.3.1, we found that the expense to calculate the spatial index number using scheme 2 is simpler and faster than using scheme 1. It is because there is no need to calculate the relative location in map scales 1:20K, 1:10K and 1:5K using scheme 2, which is a substantial reduction of calculation cost in mobile device.

Empirically, a test is conducted in a mobile device against two sets of small tile images database, which is created using the scheme 1 and scheme 2 algorithms. The test results revealed that the retrieval of a record from scheme 2 table is faster than from scheme 1. The difference is ranging from 50 to 100% faster.

3. AN ADAPTIVE APPROACH TO TRANSFERRING GEOSPATIAL INFORMATION

The proposed algorithm is trying to deliver geospatial data in a systemic approach after considering the client's situation and need such as speed of client (4; 50; 70 Km/h), and moving direction (e.g., North; South; East; West). The primary idea is to partition the features geographically into a number of blocks according to the client's allowable display area and send the blocks asynchronously to the client. The blocks of features received by the client will form a dynamic moving local geospatial database on the client device. Thus an AJAX-like agent is developed to implement on the client application tier and is responsible for both interacting with the client user and communicating with mobile GIS middleware on behalf of the user.

Basically this simple approach is employing to ensure a continuous graphic area comprised of at most eleven by eleven tiles of map image is provided on the client's device for viewing as if it were seamless.

3.1 Assembling of Base Block of Objects

There are two variants for assembling the base block of objects (BBO) to form part of a dynamic moving local geospatial database: (1) even number of block assembling method, and (2) odd number of block assembling method.

3.1.1 Even number of block assembling method

The dynamic moving local database is a square block of BBO producing by assembling even number of rows and the same

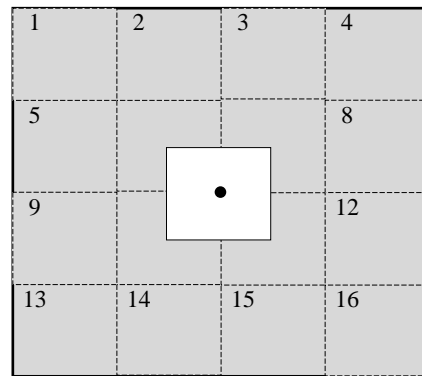


Figure 7. Assembling of 4 rows by 4 columns of BBO

even number of columns, e.g. 4x4, 6x6, 8x8, etc.

Take a 4x4 assembling for illustration. In the 4x4 assembling method is a combination of 16 blocks of features. The centre of the display area (same size as BBO) is positioned at the lower right corner of block 6. In other word, the display area of the device will cover a quarter of each of the blocks 6, 7, 10 and 11 as shown in the following diagram (Figure 7).

The payload for building the initial geospatial database is the transmission of 16 blocks of BBO features. The number of blocks to be transferred in the subsequent move is depending on the updated position of the device's centre as follows:

- Position 6, 7, 10, or 11 – no block to be downloaded
- Position 2, 3, 5, 8, 9, 12, 14, or 15 – 4 blocks of features
- Position 1, 4, 13, or 16 – 7 blocks of features

3.1.2 Odd number of block assembling method

Again, this assembling method will guarantee a local dynamic moving geospatial database containing a square block of BBO features is available on the mobile client device at any time. This assembling method will guarantee the viewing area is exactly one of the constituent BBO located at the centre of the larger block. Take a 5x5 block as an example, block number 13 will be displayed on the device screen and containing the client current location as shown in Figure 8.

Again, the volume of data to be transferred is depending on two situations in order to maintain a 25-block local database.

- **Initial stage.** It is necessary to send 25 blocks of features to the client in the beginning. The transfer must start with block 13 and then follow by inner ring blocks (i.e., blocks 8, 12, 14, 18, 7, 9, 17, and 19). Finally, the remaining outer ring blocks.
- **Subsequent move.** Moving to positions 8, 12, 14, or 18 will trigger the transmission of 5 blocks of features. Whereas, moving to positions 7, 9, 17, and 19 will trigger the transmission of 9 blocks of features.

3.2 Total Number of Blocks Transferred to Form the BBO

Typically, the volume of blocks to be transferred to form the local database is depending on two situations.

- **Initial formation.** The total number of blocks to be transferred to form the BBO is $n \times n$ (where n is the width of the BBO).
- **Subsequent update of database.** The total number of blocks to be transferred to update the BBO is (i) n for movement directions in North, South, East and West; and (ii) $2 \times n - 1$ for movement directions in NW, NE, SW and SE.

3.3 Assembling of Rectangular Buffer Strip

Rectangular buffer strip is part of the local dynamic moving database. It is the outer region of the square block of BBO. The size of this strip is varying and default to a strip of 1 block around the square block of BBO.

Typically, the total number of blocks to form the rectangular buffer strip is given by the following equation.

$$B_{BBO} = 4s(n + s) \tag{11}$$

Where s is the strip size in block value, and n is the width of BBO in block value.

3.4 Formation of Dynamic Moving Geospatial Database

The flow of the information is carried out in two stages: (1) Initial formation of local geospatial database; and (2) Updating of local geospatial database.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Figure 8. A block of 5x5 BBOs

3.4.1 Initial formation of dynamic moving database

The initial formation of dynamic moving database involves the formation of BBO and RBO. The formation of BBO and RBO is the process of requesting constituent blocks from the master database, which can be resided locally on the device or remotely on a server.

The total number of blocks to be transferred is the sum of blocks for BBO and RBO as given by the following equation:

$$B_{LDB} = n^2 + 4s(n + s) = (n + 2s)^2 \tag{12}$$

The SQL select statement to extract the relevant blocks for BBO and RBO from the master database is listed as follows:

```
SELECT PKUID_s2, TileImage
```

42	43	44	45	46	47	48
35	36	37	38	39	40	41
28	29	30	31	32	33	34
21	22	23	24	25	26	27
14	15	16	17	18	19	20
7	8	9	10	11	12	13
0	1	2	3	4	5	6

Figure 9. Arrangement for 1-dimensional array storing IDs of BBO

23	24	25	26	27	28	29	30	31
21								22
19								20
17								18
15								16
13								14
11								12
9								10
0	1	2	3	4	5	6	7	8

Figure 10. Arrangement for 1-dimensional array storing IDs of RBO

```
FROM basemap_image_06071011
WHERE PKUID_s2 IN ( list_of_ids )
ORDER BY PKUID_s2 ASC
;
```

Since we are employing the scheme 2 spatial index numbering method, there are two characteristic for this scheme: (1) same row of blocks will have the smallest ID located on the left hand side and will have the largest ID located on the right hand side; (2) the ID for a block in the higher row is larger than the ID in the lower row.

Based on the characteristics of this scheme, we can express the ID information and their spatial relationship with two 1-dimensional arrays for BBO and RBO as illustrated in Figures 9 and 10.

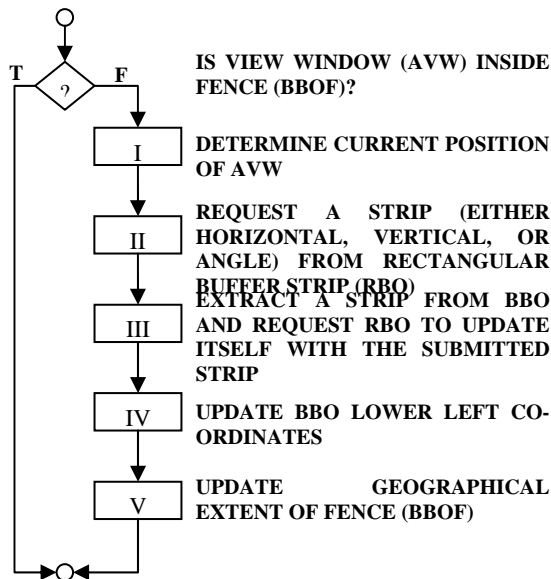


FIGURE 11. ALGORITHM FOR GENERIC UPDATING OF LOCAL DB

The ID list provided to the SQL select statement is sorted in ascending order according to the characteristics of this scheme.

3.5 Generic Updating Process on Local Dynamic Database

The local dynamic database is needed to go through an updating process when the centre of the view window is touching or exceeding the fence boundary. The generic algorithm for updating local database is carried out as shown in Figure 11. The RBO updating process as mentioned in Task III of the algorithm is a separate process that needs interaction with the master geospatial database to fetch new base blocks. The detailed updating process is varying depending on the relative position of the view window with respect to the fence boundary.

4. IMPLEMENTATION OF THE PROPOSED APPROACH ON A WINDOWS MOBILE-BASED DEVICE

The development of a Windows Mobile-based application called amGIS.Viewer.SA is employing all the concepts and algorithms described earlier. The application is using a set of DLLs customised for Windows Mobile device as illustrated in Shea and Cao (2010) as the building block for application development.

Figures 12 through 14 illustrated the execution of this application in a Dell Axim X51V pocket PC running Windows Mobile 5.0.

5. CONCLUSIONS

We proposed an algorithm to define a unique spatial index numbering system for fast retrieval of geospatial data under standard RDBMS environment on a Windows Mobile-based device without using R-Tree algorithm. The benefit of applying this unique spatial index number in fetching geospatial data over the use of R-Tree algorithm is faster retrieval of result. It is because the expensive computing cost of R-Tree is not required.

REFERENCES

- Cantor, G., 1845-1918. Transfinite Numbers and Set Theory. <http://www.math.utah.edu/~pa/math/sets.html> (accessed 18 Aug. 2009)
- Shea, G.Y.K., Cao, J.N., 2010. Use of open source programs to create a foundation for developing serious GIS application on mobile device. Proceedings of the XXIV FIG International Congress, 11-16 April 2010, Sydney, Australia.

ACKNOWLEDGEMENTS

Acknowledgement is due to The Hong Kong Polytechnic University for the financial assistance of this study. Thanks also go 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) for this project.

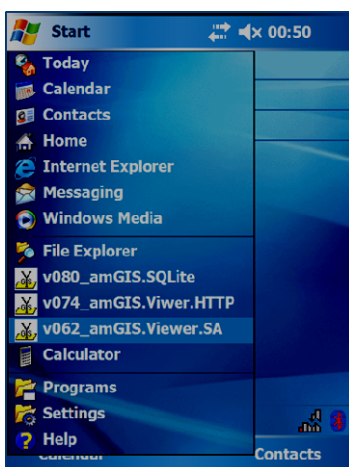


Figure 12. Start amGIS.Viewer.SA from the Start Menu

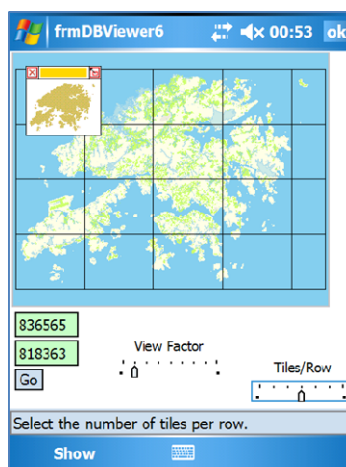


Figure 13. Select the number of tiles per row and select a point

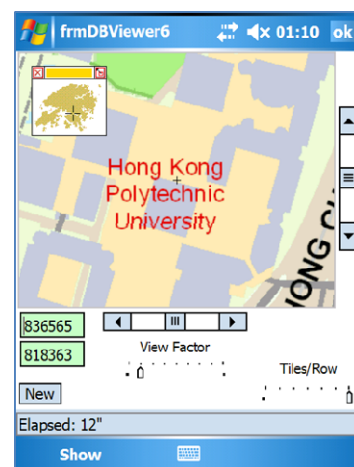


Figure 14. Centre tile of the 9x9 block of image is shown