© 2015 IJSRSET | Volume 1 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

# Lossless Compression for Raster Maps Based On Mobile

**Phyo Phyo Wai*1, Nyein Aye2, Mie Mie Khin2**
*1University of Computer Studies, Mandalay, Myanmar
2Hardware Department, University of Computer Studies, Mandalay, Myanmar
3Pro-Rector, University of Computer Studies, Mandalay, Myanmar

## ABSTRACT

In this paper, the combination of quadtree and block encoding compression approach is proposed for lossless compression of raster maps. This application is implemented on the mobile phone. Firstly, this approach iteratively split the raster map into quarter sections and the quadtree structure will construct. To index the spatial data, quadtree uses Morton encoding method. This approach defined a limit of leaf level of quadtree. When the splitting is reach to the limit level, each leaf nodes use the block encoding approach.
**Keywords:** Quadtree, Block Encoding, Raster Map, Lossless Compression

## I. INTRODUCTION

With the growth of using portable devices such as mobile phones, using maps in the phones are become popular. The maps are useful to know the current locations of the user who want to know their location from their devices when connecting the Internet. The original sizes of the maps are very extreme large. So, the transfer time of the map on the phone need to be effective and the transfer map's size also need to be compress. Raster maps which one the data type of Geographical Information System (GIS) are based on the pixels of grid cell. One pixel can be a very importance place of earth. So, at the time of compress the maps need to be lossless. It is very important. When some pixels are lost at the compress time, the information on the map will also be missed information. Therefore, lossless compressions of raster maps are essential. When compressing the maps, many lossless compression techniques can use.

A raster map is an image generated from a vector representation of a map. A raster map is comparatively data-redundant, but has a simple data structure, and can be easily visualized. Currently, vector maps and raster maps usually coexist in the navigation systems. In fact, main commercial navigation systems, e.g., Google Map and Baidu Map, provide only raster maps to the clients and thus protect their vector data from running into their competitors. When transmitting a raster map over the wireless network, the data size is expected to be the smaller the better [1].

Technically speaking a raster image is a two-dimensional rectangular matrix of individual pixels. Each pixel corresponds (directly or indirectly) to some well defined color. And colors are defined as RGB values, i.e. as a triplet of values indicating the Red, Green and Blue relative intensity. Most usually an 8-bit color depth is used, and so a 0 value corresponds to "completely off", i.e. black, and 255 to "completely on", i.e. pure red or pure green or pure blue. Any real color can be encoded using an appropriate RGB value: an 8-bit color depth allows RGB to represent 16,777,216 different colors, i.e. the so called true color commonly used by digital cameras, screens, color printers and other well known digital imagery equipments. This is also known as the RGB color space (commonly used on color digital photography): each pixel requires 3 bytes, i.e. 24 bits to be represented into this color space [11]. With the development of GIS technology, the compression techniques of image data are very important. Raster image data are heavy size. So, it needs to compress. Large amounts of data can create enormous problems in storage space and transmission time. The main reasons of data compression could be summarized as: 1) multimedia data have large data volume and 2) difficulty sending real-time uncompressed data over current network.

Compression data principle is to eliminate data redundancy and try to find a code with less data volume. All image compression techniques try to get rid of the inherent redundancy, which may be spatial (neighboring similarity or equal pixels), spectral (pixels in different spectral bands in a color image) or temporal (correlated images in a sequence, e.g. television) [2].

Two compression techniques are mostly used: Lossless and Lossy compression. A lossless compression algorithm eliminates only redundant information, so that one can recover the data exactly upon decompression of the file. Lossless data compression is compression without any loss of data quality. The decompressed file is an exact replica of the original one. Lossless compression is used when it is important that the original and the decompressed data be identical. It is done by re-writing the data in a more space efficient way, removing all kinds of repetitions (compression ratio 2:1) [3].

Quadtree is a class of hierarchical data structures which contains two types of nodes: non-leaf node and leaf node, or you can call them internal node and external node. Owing to its "Divide and Conquer" strategy, it is most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions [4]. A quadtree compression of a bi-level image is based on the principle of image compression which states; if we select a pixel in the image at random, there is a good chance that its immediate neighbors will have the same or similar color. The quadtree method scans the bitmap, area by area, looking for areas composed of identical pixels (uniform areas) [5].

Block encoding is a little like a two-dimensional version of run-length encoding in which the array is defined as a series of square blocks of the large size possible. Recursively, the array is divided using blocks of smaller and smaller size. It is sometimes described as a quadtree data structure [6].

Quadtree compression technique is very popular and quadtree has very effective segmentation result. After segmenting the raster image by using quadtree method, it will get the blocks of map. These blocks may be compress by using block encoding compression method. Block encoding method is working pixel by pixel. So, it will get lossless image when decompress. Block encoding compression technique has good compression ratio but long processing time. Quadtree compression technique has less processing time but poor compression ratio. So, by doing hybrid of these two compression techniques, the proposed system will get good compression ratio, effective processing time and lossless raster map that identical with original map.

The organization of this document is as follows. In Section 2 describes the proposed raster map lossless compression approach in details. In Section 3 presents the experimental results and in Section 4 concludes our work.

## II. METHODS AND MATERIAL

### A. System Design Overview

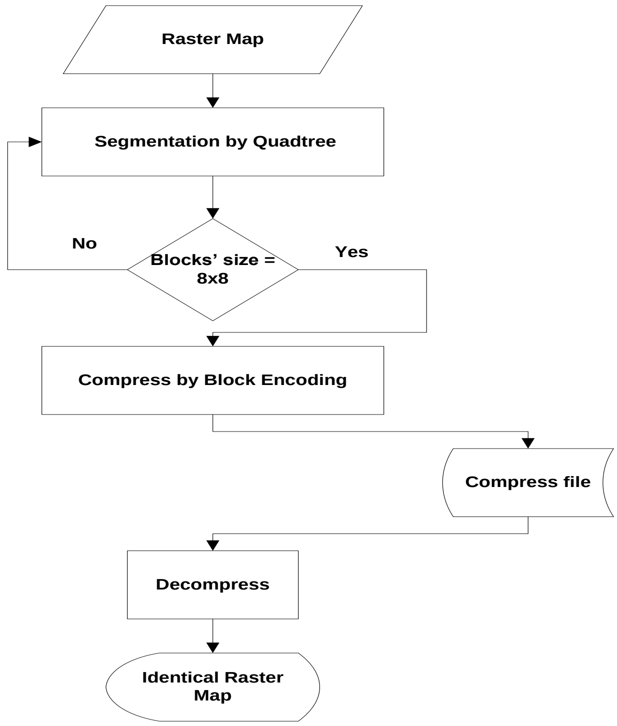



Figure 1: System Overview Design

In this system, the input data may be raster map. And then, the compression operation is start. In order to obtain the initial segmentation results of the image, the image will first iteratively split into quarter sections and the quadtree structure will construct. A spatial indexing structure will built using improved Morton encoding based on this structure, which provides the merging process with data structure for neighborhood queries. In quadtree structure, the image's size of width and height must be the same length. It will divide the image until it reaches the size of 8 by 8 pixels in leaf node. At the time of the image is reached the limit of size, the block encoding method will be use for each block. The block-encoding method has neighbouring boundary pixel problem and the quadtree coding method has multi-tree level problem. By embedding block-encoding method in quadtree coding method, we will solve these problems, effective the processing time and get high accuracy images results. In next step, it saves the compress image file. And then decompress the image file and show the image like originally.

### B. Quadtree Compression Approache for the System

In general, our quadtree method has two major steps: segmentation step and a bottom-up color pixel region merging step. In the segmentation step, a quadtree initial segmentation is performed first, providing the color pixel merging process with basic color elements. In this step, we also conduct a spatial indexing creation and color pixel calculation, in order to provide color values and neighborhood information. In the second step, we use color table to express the relationship between color pixels. In color table, regions are represented by nodes. Region merging is performed between the most similar and adjacent color pixels. Region merging will stop until the last layer is minimum block size 8. After the merging process, the final segmentation result is obtained. Figure 2 shows the procedure of our quadtree compression method.

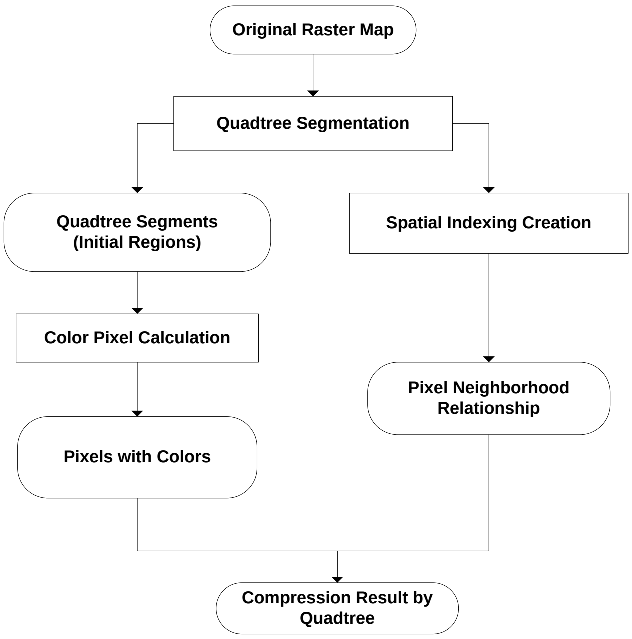



Figure 2: Quadtree Processes of the System

1) Quadtree Segmentation**:** The entire image is treated as quadtree root, and then the iterative quadtree segmentation is conducted for each sub-region until all regions have satisfied the given criterion threshold. Quadtree segmentation splits images into four parts of the same size. However, for the image with a large length-width ratio (e.g., image strip along an imaging orbit), this method will produce narrow strip initial segments. So, before the segmentation, we will judge the image length-width ratio. According to experience, we use '0' as the length-width ratio threshold. When the ratio is larger than '0', the image will be cut along the longer edge direction, in order to make the length-width ratio for each slide less than 0. Then, we perform quadtree segmentation on each image slide independently, evading narrow strip segmentation. The following study focuses on images with a length-width ratio of less than '0'.
2) Color Pixel Calculation: First of all, we measure the color data, values of x and y, and width and height of the image.
3) Spatial Indexing Based on Improved Morton Coding: Region merging is performed between neighbouring nodes. Because segments generated from quadtree segmentation have different sizes, it is difficult to perform a quadtree neighbourhood searching between segments. Direct node traversal requires a great computation. Hence, an indexing mechanism for efficient neighbourhood searching needs to be constructed. We use Morton coding [8, 9] as our basic method, and improve its efficiency, in order to make it more effective when applied to huge image segmentation.

Morton coding assumes that the code of the root node (represents the whole image) is 0. For each of the inner nodes encoding as $i(i \geq 0)$. Codes of its 4 children are $4i + k$ ($k$ = 1,2,3,4). Through this coding mechanism, the layer index of node $i$ can be obtained by $[\log4(3i + 1)]$ ($i \geq 0$) and its parent code can be obtained by $[(i - 1)/4]$ ($i >$ 0). Figure 3 is an example of quadtree segmentation and its Morton coding.

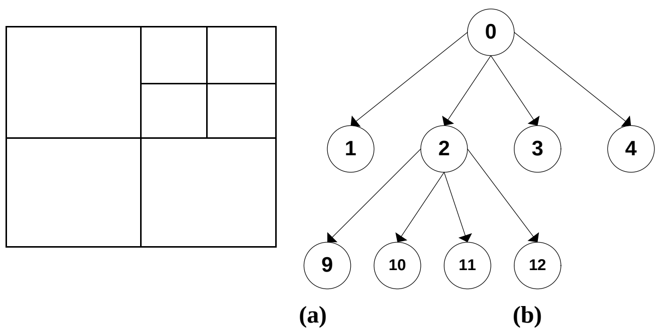



Figure 3: Quadtree segmentation and Morton coding (a) Image quadtree segmentation; (b) Corresponding quadtree with Morton coding.

Coding each cell in the uniform grid by using Morton code $M_{i,j} = M'_{i,j} + \sum_{i=0}^{i<n} 4^i$, where $n \geq 0$ is the total layer of the quadtree. $M_{i,j}$ is the binary interleave coding of row index $i$ and column index $j$, in the decimal representation. Take location at row 3 and column 2 as an example. Binary expressions of row and column are, 3 = 0011 and 2 = 0010, respectively. So, the binary interleave is 00001110, and its decimal value is 14. Therefore, its Morton code is $14 + \sum_{i=0}^{i<2} 4^i = 19$. See the right schematic in Figure 3(b) for detail of constructing binary interleaves.

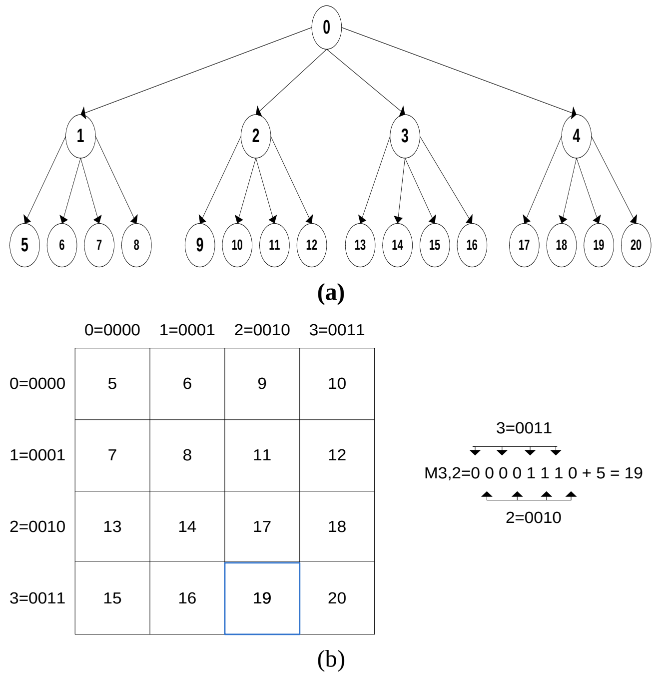



Figure 4: Complete uniform grid and the Morton coding of the example shown in Figure 4. (a) Tree structure of the example shown in Figure 4 and (b) the corresponding uniform grid and Morton coding.

### C. Block Encoding Methodology of the System

In block encoding method of this system, the input data is from 8x8 pixel raster map of quadtree segmenting compression result. In our block encoding method consists of two parts, namely flag encoding and block data encoding. Flag encoding records the location and color information of the block targets, and block data encoding records get block data and their boundary information. The general procedure of block encoding method for this system is show in the following Figure.

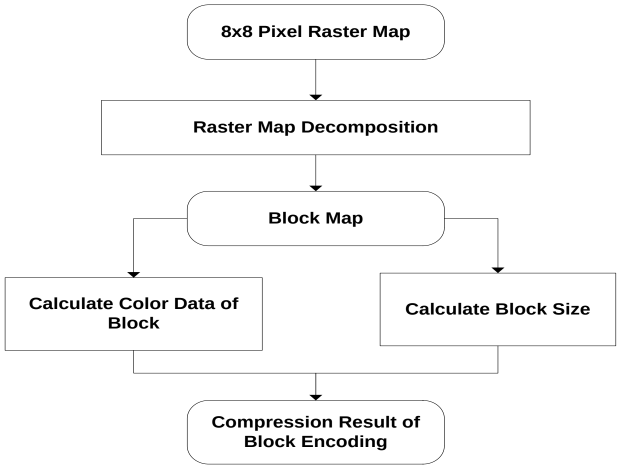



Figure 5: Block Encoding Process of the System

1) Decomposing Raster Map In raster map decomposition, first of all, we divide the 8x8 raster map into four pieces of 2x2 square blocks.
2) Flag Encoding: Since pixels in one separate block share the same RGB value, flag encoding achieves the color encoding of the entire block by recording a representative pixel within the block. Since the representative pixel can be any pixel in the block, we get it from a random selection. Figure 5 shows the codeword structure of flag encoding, where the first(row + col) bits record the coordinates of the representative pixel and the next 24 bits record the color.The problem remained is to decide the value of i and j. Suppose the size of the raster map is X × Y, the horizontal and vertical coordinates of any pixel can be stored by row bits and col bits, where row and col are constraint by Eq. (3).

$$\begin{cases} 2^{row-1} < X < 2^{row} \\ 2^{col-1} < Y < 2^{col} \end{cases}$$

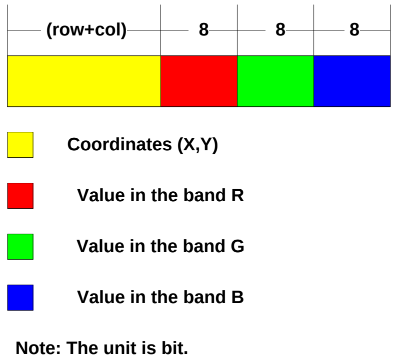



Figure 6: The Flag Encoding Codeword Structure

In order to introduce block encoding process in details, we take Figure as an example. According to the definition of block targets, there are four different block targets in Figure, marked as **I**, **II**, **III** and **IV**. The location and color information of block **I** could refer to the point $M$. The coordinates of $M$ are (3, 3) and the color is RGB(255,255,0). Therefore, the flag encoding of the block **I** is as follows:

$$\underbrace{0\ldots\ldots011}_{row\text{ bit}}\ \underbrace{0\ldots\ldots011}_{col\text{ bit}}\ \underbrace{1111\ 1111}_{8\text{ bit}}\ \underbrace{1111\ 1111}_{8\text{ bit}}\ \underbrace{0000\ 0000}_{8\text{ bit}}$$

These data results of row, column and color values are use in the color matrix.

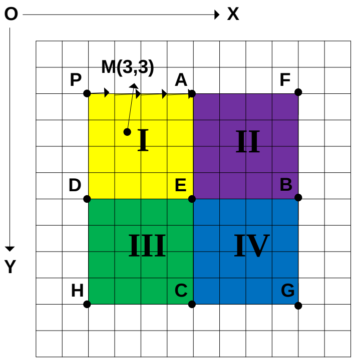



Figure 7: A Block Encoding Example

3) Block Data Encoding: In block data encoding, we do two steps. First step is getting the block data and second step is calculating the block size. In get block data, we get the same color block size from calculate block size. Block data will get xNode + row, yNode + clm and same color block size. In calculate block size, we calculating the pixel value with pixel by pixel based row order by intending the lossless. So, we use the idea of row by row raster scan order. In this calculation, the main intended area is first row and first column of block and last row and last column of image. Because we need to know the beginning of block that remark as a start region of block, and we need to know the end of the image that to stop the indexing.

### D. Combination of Quadtree and Block Encoding

At this step, the output result map of the quadtree will be the input map of the block encoding method. Early, we considered to compress the image by using block encoding and then we will use the quadtree. But, block encoding has boundary tree problem and block encoding is indexing the image pixel by pixel. So, compressing time is long. Its data structure is likely quadtree data structure. And, quadtree segmentation is time effective and morton spatial indexing structure is also the effect for quadtree. So, we divide the image using the quadtree first of all. The problem of quadtree is layering level node problem. When nodes are overloading, searching to root is difficult. So, in level detail step of quadtree, we use block encoding. At the time of combing these two methods, we considered to use histogram first. But, histogram measuring can't get pixel color detail. So we define the minimum block size as 8x8. By defining these minimum block size, we can reduce the quadtree levels and indexing pixel by pixel time of block encoding can effective. Moreover, we can produce the output lossless raster map that identical with the original map.

## III. RESULTS AND DISCUSSION

### A. a **Dataset**

A collection of 65 raster-map images [1, 10] has been collected in this study. Each image is originally generated from a vector representation of a map. All images in the dataset share the size of 1024×1024, 800x800, and are stored in BMP format with 24- bits true color. Ten maps of them are shown in Figure 9.

### B. **Overall Performance**

To evaluate the performance of the proposed system, we compute the compression ratios for the 69 images in the dataset by using following equation [5].

$$Compression\ Ratio = \frac{size\ of\ the\ output\ stream}{size\ of\ the\ input\ stream}$$

We calculate the compression ratio of one image from Map Set #1 by choosing randomly. In this image, original size is 1024 x 1024 pixel size bitmap file. Original file size is 3MB. The output file size after compressing is 1.3MB. When calculating the compression ratio, the compression ratio output is 0.4. And, it is calculating by using this equation 100 x (1-compression ratio) for a reasonable measure of compression performance. When inserting the compression ratio into the equation, 100 x ( 1 – 0.4) = 60 it get the output result is 60. A value of 60 means that the output stream occupies 40% of its original size (or that the compression has resulted in savings of 60%). In this system time is not effective as much as run in the computer. Because this system is run on the phone that has the OS version is v4.4.2 (KitKat), Chipset is Qualcomm MSM8226 Snapdragon 400, CPU is Quad-core 1.2 GHz Cortex-A7, GPU is Adreno 305, internal memory storage is 8 GB and RAM is 1.5 GB.

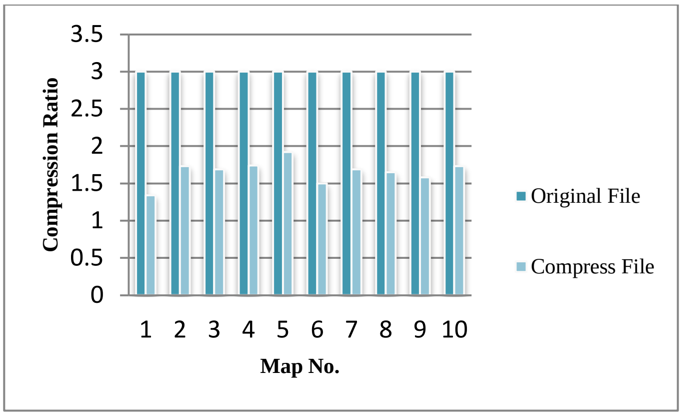
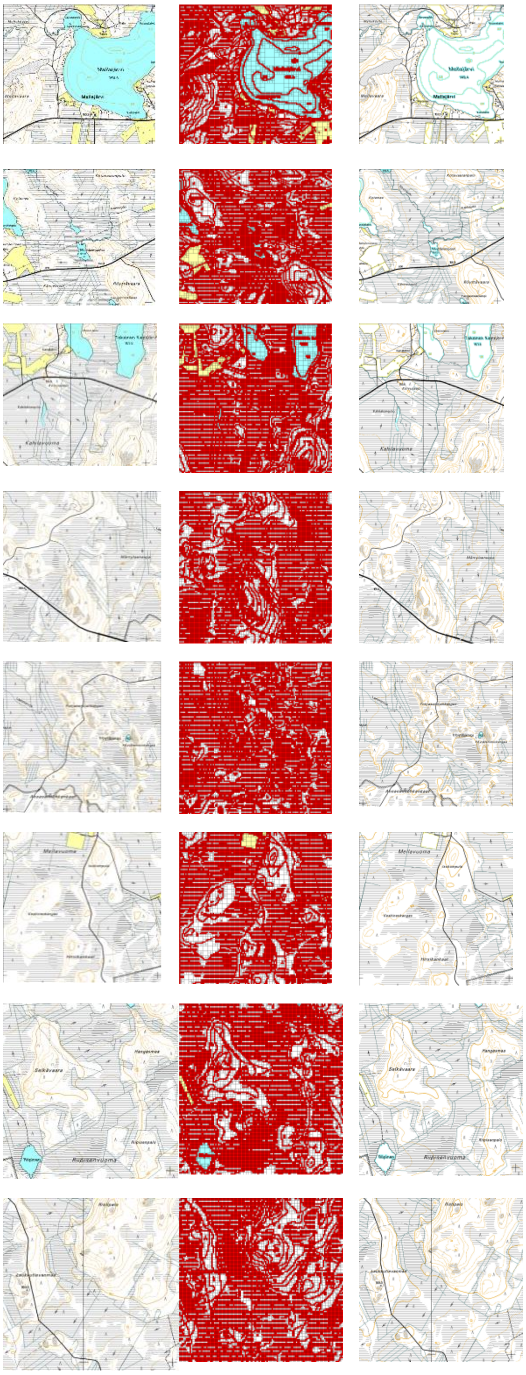



Figure 8: The Chart of 10 images



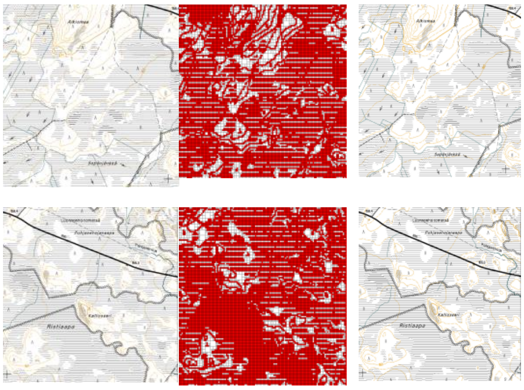


Figure 9: The randomly 10 maps. Column 1 original map, column 2 quadtree map, column 3 block encoding map

## IV. CONCLUSION

The compression of image is save the storage space and it is effective the sending data over current network. The proposed system uses block encoding method and quadtree method. This proposed system will more suitable and more compress than other compression idea.

## V. REFERENCES


[1] D.J. Abel and J.L. Smith, "A data structure and algorithm based on a linear key for a rectangle retrieval problem", Computer Vision, Graphics, and Image Processing 24, 1(October 1983), 1-13.

[2] H. Samet, "Hierarchical representations of collections of small rectangles", ACM Computing Surveys £0, 4(December 1988), 271-309.

[3] Claudia Dolci, Dante Salvini, Michael Schrattner, Robert Weibel, "Structures for Data Compression".

[4] Xiang Yin, "Quadtree Representation & Compression of Spatial Data".

[5] David Salomon, Giovanni Motta," Handbook of Data Compression", 5th ed. DOI 10.1007/978-1-84882-903-9_7, © Springer-Verlag London Limited 2010

[6] Paul Longley, "Geographic Information Systems and Science", John Wiley & Sons, Mar 22, 2005 - Science - 517 pages

[7] Claudia Dolci, Dante Salvini, Michael Schrattner, Robert Weibel, "Structures for Data Compression".

[8] Gang Fu, Hongrui Zhao, Cong Li and Limei Shi, "Segmentation for High-Resolution Optical Remote Sensing Imagery Using Improved Quadtree and Region Adjacency Graph Technique", Remote Sens. 2013, 5, 3259-3279; doi:10.3390/rs5073259

[9] D.J. Abel and J.L. Smith, "A data structure and algorithm based on a linear key for a rectangle retrieval problem", Computer Vision, Graphics, and Image Processing 24, 1(October 1983), 1-13.

[10] http://cs.joensuu.fi/sipu/images/

[11] www.gaia-gis.it/spatialite/rasterlite-man.pdf.