A lossless coding scheme for maps using binary wavelet transform

Angel Caroline Johnsy and Gilda Schirinzi

Dipartimento di Ingegneria, Università degli Studi di Napoli “Parthenope”, Napoli, Italy

ABSTRACT

The maps and images of geographical information system (GIS) are used for finding locations, accessing rail, bus routes, and for educational purposes such as study on vegetation, landscapes, population, and so on and so forth. Remote sensing is the process of acquiring data about Earth by using satellites or satellite-borne or airborne sensors. The images acquired through remote sensing systems are integrated within GIS to store, analyze, and manipulate geographical information of the Earth. The huge size of the digital raster maps makes compression inevitable in particular to reduce the transmission time and display them on the Internet as well as other networks. In this paper, a lossless coding approach that performs encoding on the decomposed binary layers by taking the advantage of binary wavelet transform that produces sparse matrix for row column reduction and Huffman coding is presented. The results obtained on raster maps are compared with those of other existing techniques.

Introduction

Images obtained from satellites and other airborne imaging systems are often referred as remote sensing data (Liu and Mason 2009). Remote sensing can be used to collect raster data maps for further processing in different bands of the electromagnetic spectrum for identification of various objects of interest and to monitor their changes. These remote sensed data are used by various government mapping agencies and software developers to create various products, such as digital raster graphic (topographic maps), digital elevation model, etc. The geographical information system (GIS) is the one that stores, manipulates, and analyses the remote sensed data (Campbell 2011). The GIS file format is a standard of encoding geographical information into a file. A broad classification of the file formats includes raster, vectors, and grid formats. Topographic maps are created using photogrammetric interpretation of various images obtained from satellites, LiDAR, or other remote sensing systems. These maps comprise various products such as roads, forest cover, urban areas, contours, streams, lakes, etc. Raster files are in particular used for storing remote sensing data. These are also used to store image information like scanned paper maps and aerial pictures. Digital maps can be stored and distributed using compressed raster image formats or rectangular array of pixels. Raster data type consists of rows and columns of cells, with each cell storing a single value. Raster data can be images with each pixel containing a color value and are stored in various formats, such as standard file-based structures (like TIF, JPEG, or Binary Large Object (BLOB) data). The raster format is preferable due to the simplicity of its direct access. The United States Geological Survey (USGS), a civilian federal agency, produces several national series of topographic maps which vary in scale and extent. For instance, a 1:50,000 scale topographic map or its equivalent is produced for states like Michigan, California, etc. These large scale topo maps give rise to the concern of storage space and also transmission through the network. For compression of graphics and charts, lossless compression method is usually preferred because of the need to retrieve accurate geographical locations for further analysis or derive new data and produce a by-product from the raster map. Lossless data compression is a class of data compression algorithms that allows the original data to be reconstructed from the compressed data without any loss. The GIS Image compression system for large maps supports compact storage size and decompression of the image. ArcGIS (Esri 1999), a GIS for creating and processing maps performs JPEG 2000 or LZ77 (Ziv and Lempel 1977) compression for lossless compression on raster maps. Adopting a lossless compression scheme, which can compress data with less computational complexity and also save space, is inevitable for maps that occupy huge storage space. In Alzahr and Borici (2015), a lossless compression technique mainly for bi-level image or color maps, row column (RC) reduction technique is proposed to show that...
the maximum compression achieved on an 8 × 8 block (64 bits) is 17 bits. The coding technique involves construction of a Row vector (8 bits), a Column vector (8 bits), and then the reduced block. The blocks that are not reducible using this method will be stored as 64 bits. A codebook is constructed for the frequently occurring blocks by finding the probability of occurrence based on Huffman coding and utilized during compression to skip the RC coding on blocks. In Swanson and Tewfik (1996), a binary wavelet decomposition on binary images is proposed which uses simple mod-2 (modulo-2) operations. This approach is an alternate to the real wavelet transformation in binary domain. In Pan et al. (2007), an in-place binary implementation is discussed for gray scale images. A gray scale image undergoes bit plane coding and the bi-level layers are extracted starting from most significant bit to least significant bit or vice versa to exploit the advantage of sparsity after the Binary Wavelet Transformation (BWT). In Gilmudtinov et al. (2014), the lossless coding is done on images by performing binary layer decomposition, which follows predictive coding to achieve compression.

In this work, we propose a lossless compression technique for huge images in particular raster maps with discrete colors. Based on the USGS standards, the color palette is limited to 13 colors to represent various areas such as road, water bases, vegetation, boundaries, contours, and revised areas. This limitation in colors has advantage in compression due to the large size of the maps. The decomposition of the image into binary layers makes the data bi-level so that they can be partitioned into 8 × 8 blocks and encoded. Since the image is already sparse to certain extent due to decomposition, the 8 × 8 block undergoes RC reduction which reduces the block else a BWT that can produce sparse block is done followed by Huffman coding.

**Methods Adopted**

**Row Column Reduction**

RC coding is an approach in lossless coding scheme for raster maps. For an 8 × 8 block, a Row reference vector, a Column reference vector, and a reduced matrix is generated. The Row reference vector is constructed by creating an 8 × 1 empty vector. The subtraction of subsequent rows from the 8 × 8 block is done and the sum of each row is stored in the Row reference vector. Reduce the 8 × 8 block using the Row reference vector. The Column reference vector is constructed by creating a 1 × 8 empty vector. The subtraction of subsequent columns from the reduced block is done and the sum of each column is stored in the Column reference vector. With this approach, the maximum compression ratio on a block will be 3.76, since the minimum length of the reduced block is 17 bits; 1 bit from the block, a Row vector (8 bits) and a Column vector (8 bits). The 8 × 8 block dimension is chosen empirically, in such a way to exploit the advantage of redundancy present in the huge maps, while maintaining low the computation time. Blocks of size greater than 8 × 8 lack redundancy that takes advantage of RC reduction (i.e. identical rows or identical columns) while for blocks of size less than 8 × 8 the computation time and storage increase drastically due to increase in number of blocks to be processed and stored.

**Binary Wavelet Transformation**

Figure 1 shows the construction of a two-band discrete orthonormal BWT. It is equivalent to the design of a two-band reconstruction filter bank with added vanishing moment conditions. In BWT, the input signal is passed in parallel through the low-pass and band-pass filters. The outputs of both are then decimated by a factor of 2 to obtain a detailed component and an approximation component of the original signal. This part of the filter is known to be Analysis Filter Bank. The two decimated signals may then be up-sampled and passed through the corresponding inverse filters. The output obtained by these two filters is summed up for reconstruction of the original signal. This is similar to the real field wavelet transform except for the fact that the original and transformed signals are reconstructed in the binary domain. An in-place implementation in the spatial domain helps greatly in reducing the computational complexity and conserves the memory storage required for the transformation in the real field wavelet transform. The binary field wavelet transform also has an in-place implementation similar to that of real field wavelet transform. This makes the transformation more efficient by both reducing computational complexity as well as utilizing lesser memory storage. Table 1 shows the filter coefficients of length 8 for binary. In this process, initially the samples are split into two sequences by their corre-

![Figure 1. Filter bank implementation of the BWT.](image-url)
sponding index positions (odd indexed samples in one sequence while even indexed samples in the other). The sequences are then updated according to the corresponding filter coefficients from low-pass and band-pass filters. This can be compared to that of a “SPLIT UPDATE AND PREDICT” procedure in the lifting scheme of the real field wavelet transform. The outputs from the low-pass and band-pass filters are then interleaved to obtain the final transformed output. In this article, we study and analyze three different filter groups for performing wavelet transformation. Figure 2 shows the in-place implementation of the binary filter of Group1 coefficients. Binary wavelet decomposition is applied on binary images using mod-2 operation which is equivalent to an exclusive OR (XOR) operation. Advantage of BWT over the other wavelet transforms is that the intermediate or the transformed results are binary. Consider the vector of size 1 × 8 as shown in the Figure 2. BWT with group 1 coefficients is performed. The odd indexed samples are unchanged and the XOR operation (mod-2 arithmetic) is performed on every even indexed sample and the neighboring odd indexed sample. Table 2 represents the forward transform for all the possible combination of 1 and 0 in binary domain. Here $x_0$ indicates even index and $x_1$ indicates the odd index.

Figure 3 depicts the integer wavelet decomposition using Haar coefficients (Strang and Nguyen 1996). The results obtained using Haar lifting scheme undergoes modulo-2 arithmetic operations to sustain the values in binary. When even index $X_0$ is zero and odd index $X_1$ is zero then the approximation coefficient $(0 + 0)/2$ is zero and the detail coefficient is zero. On application of mod-2 arithmetic on the result, it gives the approximation coefficient and detail coefficient equal to 0. When $X_0$ is 0 and $X_1$ is 1 the approximation coefficient $(0 + 1)/2 = 0$, floor off operation is performed when the even index is smaller than the odd index.

And the detail coefficient $(1 - 0) = 1$. On performing mod-2 arithmetic, the approximation and detail coefficients become 0 and 1, respectively. Similarly, when $X_0 = 1$ and $X_1 = 0$ the approximation coefficient becomes $(1 + 0)/2 = 1$, ceil of operation is performed since even index is larger than the odd index and the detail coefficient is $(0 - 1) = -1$. After mod-2 arithmetic, the result is 1 for approximation coefficient and 1 for detail coefficient. Similarly, for $x_0 = 1$ and $x_1 = 0$ then approximation coefficient is 1 and detail coefficient is zero. In order to implement in-place binary Haar wavelet lifting scheme, the results obtained using the integer Haar wavelet transform after mod-2 arithmetic operation is fed into Karnaugh map minimizer (Mano 1979) to obtain a Boolean function $f_1(x_0, x_1)$ for approximation coefficient $\frac{(x_0 + x_1)}{2}$ and another Boolean function $f_2(x_0, x_1)$ for detail coefficient is $(x_1 - x_0)$.

Table 3 depicts the binary implementation of Haar wavelet which involves simple binary addition based on mod-2 arithmetic. It is shown that the even index

---

**Table 1.** Filter coefficients of length 8 for binary.

| Analysis filter | Synthesis filter |
|-----------------|-----------------|
| Low pass, band pass | Group1 |
| 0,1,0,0,0,0,0,0 | 1,0,0,0,0,0,0,0 |

**Table 2.** Binary wavelet transform of group 1.

| Input | Analysis filter |
|-------|-----------------|
| $x_0$ | $x_1$ | $x_0 \oplus x_1$ | $x_1$ |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**Figure 2.** Binary filter of group.

**Figure 3.** Integer wavelet Haar transform.
is unchanged while the odd index undergoes XOR operation. Another approach we studied and analyzed is a post processing to Haar BWT (Figure 4) to obtain sparse matrix which is completely invertible during the inverse transform. After Haar BWT, the odd index undergoes one’s complement and XOR operation is performed with the adjacent even index $x_2$ followed by one’s complement. This procedure is adopted and carried on consecutively.

**Proposed Method**

### Decomposition and Partitioning

Based on number of layers in the map, binary images are created. Each layer is extracted from the map and converted into binary (Figure 5). Figure 10 shows the It contains six layers including the base layer (white color). Figure 11 represents the extracted layers from the map. Green color represents the vegetation, blue color represents water base, pink color represents the urban areas, and the remaining colors represent the contours. After extracting each layer, the image is made binary by assigning one to the base color and zero to the colored pixel. Each binary image will be divided into $8 \times 8$ blocks to perform the encoding.

### Index Updation

An Index vector IDV is constructed for the number of $8 \times 8$ blocks that is present in the binary layer. When the number of ones in a block corresponds to 64, then the IDV vector will be appended with zero else one. This approach will store 2 bits (10) in the place of 64 bits of zeros. Each time the IDV vector is updated with 1, the IDV0 will be updated correspondingly. When a combination of 1s and 0s is encountered for the first time, then IDV0 vector will be appended with one and an Index vector for Reduction IDVR will be created. It will be appended with one or zero accordingly followed by RC coding on the block.

### Row Column Coding

With RC coding, the minimum reduced bits achievable is 17. It consists of 1 bit from the reduced block followed by 8 bits from Row vector and 8 bits from Column vector. Based on the threshold value, the IDVR vector will be either updated with zero or one. When the reduced block after RC coding contains less number of bits than the threshold value then the IDVR vector is updated with zero otherwise with one. The threshold value either 32 or 48 is suitable since the number of bits in Row and Column vector is also to be taken into consideration. Consider the block in the Figure 6a. To reduce the block, construct an $8 \times 1$ Row vector and update it based on the identical adjacent rows. In the considered block, the first row is unique so update the Row vector (1) with 1, the second row and third is also unique so update the Row vector (2) and Row vector (3) with 1. The fourth row is not unique; it is identical to the previous row. So, Row vector (4) is updated with zero. Fifth row is unique, so update Row vector (5) with 1, the remaining rows are similar to the fifth row so update the Row vector from index 6 to 8 with zeros. Remove the corresponding rows from the block based on Row vector (Figure 6b) that has zeros.

Figure 7a represents the reduced block; the reduced block contains eight columns and four rows. A Column vector of size $1 \times 8$ is constructed and updated same as the previous row reduction but the traversal will be done in the column-wise rather than row-wise like in the previous case. A Column vector of size $1 \times 8$ is constructed and updated in the same way as the previous

---

**Table 3. Binary wavelet transform of Haar.**

| Input | Analysis filter |
|-------|-----------------|
| $x_0$ | $f(x_0, x_1) = x_0$ |
| $x_1$ | $f(x_0, x_1) = x_0 \oplus x_1$ |

| 0 0 | 0 0 |
|-----|-----|
| 0 1 | 0 1 |
| 1 0 | 1 0 |
| 1 1 | 1 1 |

---

**Figure 4.** Post processing on Haar.

**Figure 5.** Decomposition into binary layers.

**Figure 6.** (a) $8 \times 8$ block (b) Row vector.
row reduction but the traversal will be done in the column-wise instead of row-wise in the previous case. Thus, the $8 \times 8$ block is reduced to $4 \times 6$ block (Figure 7b) based on the Column vector shown in Figure 7c.

**Binary Wavelet Transform**

The IDVW vector for BWT will be created and updated when the IDVR vector encounters a value equal to 1 at the beginning. BWT is applied on the block and the transformed block can be reduced using RC coding, if the reduced block size is less than the threshold value else the transformed data will undergo the final compression stage. Based on the threshold value, the IDVW vector will be updated with zero or one. BWT on a block increases the number of zeros or transforms the block into a suitable format for RC coding in our case. Consider the block in the Figure 8a which is not reducible using RC coding thus undergoes BWT using group 1 filter coefficients (Figure 2). First, the transformation is done row-wise. The resulting block can be reduced using RC reduction. The result of RC reduction is a reduced block of size $4 \times 6$ and a Row and a Column vector. The transformation is done row-wise as shown in Figure 8b. The odd index remains unchanged whereas the even index undergoes XOR operation then the BWT is performed column-wise and the final transformed matrix is given in Figure 8c.

**Huffman Coding**

The final reduced data along with the dictionary for each layer will be stored separately which will be of greater help for layer-wise decoding. The size of the IDV vector is fixed as it depends on the number of partitioned blocks whereas the other Index vectors grow based on the extracted image layer. When an image layer contains more base layers, then codebook will be of small size. Based on the threshold value, the RC coding or BWT will be utilized equally for an efficient compression. The block diagram of the coding scheme is shown in Figure 9.

**Decompression**

Decoding is the straightforward reversal process. Each layer will independently undergo Huffman or arithmetic decoding and the process of reconstruction of the blocks will be done based on the vectors in the codebook. The vectors are traversed linearly for the decoding process.

Procedure for reconstruction is as follows:

a. When the IDV vector contains zero then an $8 \times 8$ block with ones is constructed, else if the IDV vector contains one then check the IDVR vector.

b. When the IDV0 vector contains zeros then the $8 \times 8$ block is filled with zeros, else if it contains one then check the IDVR vector.

c. When the IDVR vector contains zero then fetch the RC coded data along with the corresponding Row and Column vector and perform RC construction, else if the vector contains one then check the IDVW vector.

![Figure 7](image1.png)

**Figure 7.** (a) Reduced block after row-wise reduction (b) Reduced block after column-wise reduction (c) Column vector.

![Figure 8](image2.png)

**Figure 8.** (a) $8 \times 8$ Block (b) Binary wavelet transform – row-wise (c) Binary wavelet transformed block.
d. When the IDVW vector contains zero then perform RC construction followed by Inverse BWT and obtain the $8 \times 8$ block, else if it contains one then perform the Inverse BWT.

**Numerical Analysis**

The decomposed layers of the map of a region in California are depicted in Figure 11, along with the
original raster map in Figure 10. The layers represent the vegetated areas, urban areas, contours, streams, etc. Table 4 represents the compression ratio of the layers.

Compression ratio can be defined as the ratio between the size of the original data and the encoded data:
The comparison of the proposed technique with other lossless coding techniques such as Lempel–Ziv–Welch (LZW), JPEG2000 lossless, CCITT, and Deflate algorithm is performed and the results are presented. LZW algorithm was introduced as an enhanced version of LZ77 algorithm. CCITT is a lossless compression algorithm for bi-level images. And Deflate, a lossless compression scheme uses LZ77 and Huffman coding (Salomon 2007). It is evident from the table that the proposed method achieves better compression ratios in comparison with other established methods.

Table 5 illustrates the percentage of operations done for encoding each layer. For each layer, the percentage of $8 \times 8$ blocks with 64 ones, 64 zeros, blocks that underwent RC coding and BWT is shown. The Layer1 (Figure 11a) of the map consists of $87\%$ of $8 \times 8$ blocks comprised 64 ones and it is also evident from the Figure 11b that the layer has more of base layer. In Layer3 (Figure 11c) and Layer5 (Figure 11d), the number of $8 \times 8$ blocks comprised 64 ones is less when compared to other layers. There is always a trade-off between choosing RC coding and BWT. When the image is already sparse, the RC coding is efficient. On the other hand, if the image is dense as in the case of Layer3 (Figure 11c) then applying BWT is mandatory to have an effective and efficient coding. The dimension of the map is $9040 \times 6192$ and the total number of blocks is 874,620 and the size is 53,000 KB.

Binary entropy (MacKay 2003) is defined to measure the compactness of the signal representation

\[
H(p) = -p \log_2(p) - (1-p) \log_2(1-p) \tag{2}
\]

\(p\) is the probability of nonzero pixels in the image. The measure takes value between 0 and 1. When the probabilities of nonzero and zero pixels in the image are equal to each other, then the maximum value is obtained. When small value is observed, it indicates that there is a huge variation between the number of nonzero and zero pixels, which in turn states that the coding representation is efficient. Table 6 shows the comparison of entropy of the original and transformed blocks. The table clearly indicates that the number of nonzero pixels is considerably decreased after BWT.

**Table 4. Comparison of compression ratio with other methods.**

| Layers | Proposed method | LZW coding | JPEG 2000 lossless | CCITT | Deflate |
|--------|----------------|------------|-----------------|-------|--------|
| Layer1 | 23.86          | 15.52      | 15.23           | 17.08 | 15.74  |
| Layer2 | 110.82         | 52.59      | 77.09           | 56.49 | 55.98  |
| Layer3 | 6.75           | 5.21       | 4.83            | 6.22  | 4.96   |
| Layer4 | 22.51          | 14.89      | 14.51           | 14.4  | 15.72  |
| Layer5 | 10.79          | 9.02       | 9.2             | 9.90  | 8.27   |

**Table 5. Percentage of blocks compressed by index updation, RC coding, and BWT.**

| Layers | Blocks with ones | Blocks with zeros | Reducible blocks | BWT applied blocks |
|--------|------------------|-------------------|------------------|--------------------|
| Layer1 | 0.87             | 0.02              | 0.0056           | 0.0011             |
| Layer2 | 0.98             | 0.00              | 0.013            | 0.0015             |
| Layer3 | 0.75             | 0.25              | 0.22             | 0.025              |
| Layer4 | 0.87             | 0.042             | 0.081            | 0.0032             |
| Layer5 | 0.77             | 0.026             | 0.19             | 0.0059             |

Figure 11. (a) Layer1 – Streams and Lakes (b) Layer2 – Roads (c) Layer3 – Contours (d) Layer4 – Vegetated/Forest area (e) Layer5 – Urban areas(revised areas).
Figure 11c depicts the dense contour lines (brown color) layer of Petaluma River region in California that almost covers the map base (white color or background) and it is clearly evident from the Table 4 that the compression ratio of the Layer3 is less compared to other layers. The graph in the Figure 12 compares the compression ratio of the proposed method with the other established coding schemes and clearly depicts that the performance of the proposed method is better comparing with the other established algorithms. Of course, it is expected that achievable performance is strongly related to the image source (f. i. multi spectral, radar, etc.) and to the image features, such as texture, which can impair compression performance. Then, obtained performance has to be strictly referred to the class of images considered in the numerical experiments.

The execution time for the encoding of each layer is tabulated in Table 8. Since the maps are generally huge in size, after the decomposition of the images, they can be compressed independently by implementing parallel executions. The simulation has been done on Quad core processor at 3.60 GHz per processor and 8 GB memory.

### Conclusion

In this paper, the lossless compression technique particularly for raster maps that has discrete colors has been addressed. The compression is done by decomposing the map into layers that gives the advantage of working on bi-level images. Due to decomposition of the bi-level image, some layers become sparse to certain extent. The partitioning of the layers into blocks helps to reduce the data further when the block is dense. BWT or RC coding is performed in order to reduce the block size and finally Huffman coding is applied to the RC reduced block. The numerical results on the map show that the compression by this technique gives promising results and since the layers are compressed independently the partial reconstruction is also made possible. The performance analysis shows the overall idea about the image structure and various possibilities of reducing the data based on the segmented nature of the block. BWT is quite promising when sparse data is required. Moreover, since the in-place binary implementation is fast the computation is speeded up. The current article is restricted to only two to three possible filter coefficients. In future, the different lifting schemes for binary data can be introduced and the performance can be improved.

### References

Alzahir, S., and A. Borici. (2015). An innovative lossless compression method for discrete-color images. IEEE Transactions on Image Processing, 241, 44–56. http://dx.doi.org/10.1109/TIP.2014.2363411

Campbell, J. B. (2011). Introduction to remote sensing. New York: The Guilford Press.

Esri, (1999) - ArcGIS. [Online] Retrieved from http://www.esri.com/software/arcgis

Gilmudtinov, M., N. Egorov, and D. Novikov, (2014) Lossless image compression scheme with binary layers scanning. XIV International symposium on problems of redundancy in information and control systems (REDUNDANCY), St. Petersburg. doi: http://dx.doi.org/10.1109/RED.2014.7016703

Liu, J. G., and P. J. Mason. (2009). Essential image processing and GIS for remote sensing. Hoboken, NJ: John Wiley & Sons, Inc. http://dx.doi.org/10.1002/9781118687963

MacKay, D. J. C. (2003). Information theory, inference and learning algorithms. Cambridge: Cambridge University Press.

Mano, M. (1979). Digital logic and computer design. Upper Saddle River, NJ: Prentice Hall PTR.
Pan, H., W.-C. Siu, and N.-F. Law. (2007). Lossless image compression using binary wavelet transform. IET Image Processing, 14, 353–362. doi:10.1049/iet-ipr:20060195
Salomon, D. (2007). Data compression, the complete reference. London: Springer-Verlag. doi:10.1007/978-1-84628-603-2
Sayood, K. (2012). Introduction to data compression. San Francisco, CA: Morgan Kaufmann Publishers Inc.
Strang, G., & T. Nguyen. (1996). Wavelets and filter banks. Wellesley, MA: Wellesley-Cambridge Press.
Swanson, M. D., & A. H. Tewfik. (1996). A binary wavelet decomposition of binary images. IEEE Transactions on Image Processing, 512, 1637–7149. doi:10.1109/83.544571
Ziv, J., & A. Lempel. (1977). A universal algorithm for sequential data compression. IEEE Transactions on Information Theory, 23, 337–343. doi:10.1109/TIT.1977.1055714