Tree Based Search Algorithm for Binary Image Compression

Reetu Hooda
Dept. of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, AL 35899
rh0059@uah.edu

W. David Pan
Dept. of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, AL 35899
pand@uah.edu

***Abstract*—The demand for large-scale image data grows increasingly fast, resulting in a need for efficient image compression. This work aims at improving the lossless compression of binary images. To this end, we propose the use of a tree-based optimization algorithm, which searches for the best partitions of the input image into non-uniform blocks, and for the best combination of scan directions, with the goal of achieving more efficient compression of the resulting sequences of intervals between successive symbols of the same kind. The tree-based search algorithm searches for the best grid structure for adaptively partitioning the image into blocks of varying sizes. Extensive simulations of this search algorithm on various datasets demonstrated that we can achieve significantly higher compression on average than various standard binary image compression methods such as the JPEG 2000 and JBIG2 methods.**

***Index Terms*—Binary images, Lossless compression, Bitmaps, Hyperspectral images**

I. Introduction

In order to minimize the storage and transmission-bandwidth requirements, the need for image compression has increased drastically. This work is focused at improving the lossless compression of binary images. A binary image, also known as bi-level image, has only two possible values for each pixel - either "0" or "1" [1]. Binary image is a black and white image without color or without shades of gray. The image becomes a series of black dots on a white background. There are several good uses of such images. Binary images are used in black and white laser printer, document analysis, input/output devices, such as bilevel computer displays which can handle only binary images. Such images are also processed by banks, government agencies, credit card companies, airlines and insurance companies [2]. In many intelligence, surveillance, and reconnaissance (ISR) applications, it is desirable to detect potential targets or regions of interest (ROIs) within various types of hyperspectral imagery. These ROIs can then be used to extract statistics for classification, and for applications such as target identification and recognition [3][4].

Image Compression techniques can be classified as:

Lossless Compression: This type of compression technique allows complete reconstruction of the original data. Lossless algorithms are typically used in applications where loss of any data is not affordable.

Lossy compression: Lossy algorithms can only reconstruct an approximation of the original data. They are typically used for images and sound where some loss is often undetectable or at least acceptable [5][6].

Various studies in image and video coding show that decomposition of the original 2D signal (image) into a set of contiguous regions help in increasing the compression. This process is called *"segmentation"*. Its application includes redundancy reduction and source coding [7][8]. In general, segmentation-based coding algorithms provide higher compression when compared with traditional transform, such as discrete cosine transform (DCT), and sub band (SB) coding approaches. The partitioning paradigm used in algorithms with segmentation keeps the computational cost and complexity reasonably low and still achieve superior compression efficiency [9][10].

A natural image consists of regions of smoothness as well as regions with sudden variations. Thus, the image can be divided into regions having different ranges of perceptual importance. Typically, the frequency of content changes is considered a reasonable measure, as extra bits are allocated to highly detailed area(s) in the image [11]. Such regions are intrinsically less compressible than less detailed regions. By allocating uniform number of bits across an image, regions with fewer variations are coded with far more bits than needed. This provides motivation to examine image coder which uses variable size blocks according to the importance of the region being coded [12].

The proposed technique is based on the idea of utilizing the smoothness in portion of an image by splitting it into variable-size segments. Regions of the image dominated by frequent changes are retained as smaller blocks and the smooth segments are not chosen to be divided further. In other words, uniform regions are grouped into larger areas. Thus, decomposition of only highly changing region is performed.

The next section illustrates the proposed tree-based search algorithm.

II. Tree Based Search Algorithm

The Tree Based Search Algorithm can be broken down into the following three main steps to generate the final compressed file:

1) Full search of image sub-block.
2) Optimal tree structure in the set of all possible tree structures.
3) Two-level splitting of the original image.

A. Full Search

Full search is a repeated step in the algorithm which is used to examine the change in compression after partitioning a high frequency content region in an image. Full search algorithm is the most simplistic search method with few tradeoffs. The advantage of full search lies in the fact that we can find the absolute optimal solution. It guarantees to explore all the search paths to solve a given problem. However, its high computational complexity makes it difficult for real time implementation. Since full search on small data has low time complexity, the tree-based search algorithm uses full search on image blocks instead of using it on the whole image itself.

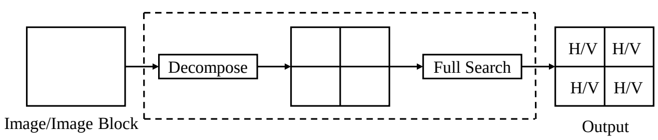


Fig. 1: Flow chart of the full search.

Fig. 1 shows the flow of full search operation used in the proposed method. Either the image or a section of it, is the input to the full search block, which divides the input into 4 equally sized blocks. Raster scan is a widely used scanning technique in image coding to order pixel values [13]. Therefore, Horizontal and Vertical scanning schemes are used to traverse the pixels of the input image block.

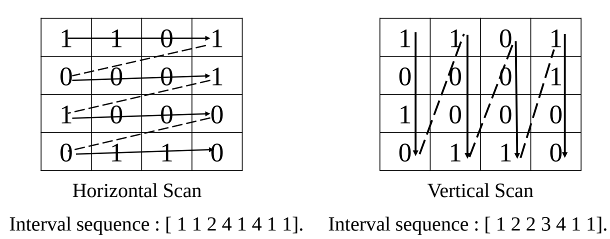


Fig. 2: Scanning patterns.

Fig. 2 illustrates horizontal and vertical raster scans to generate different interval sequences. The interval sequences generated affect the degree of compression, since the problem of compressing the original image changes to the problem of compressing the intervals as a result of a specific scan direction.

Each of the four blocks can be scanned in two directions which makes total of $2^4 = 16$ combinations to be explored. Division of the input image is followed by the full search, to examine all the possible scanning patterns and consequently finding out the best combination for highest compression. In summary, full search is used in the following operations.

1) Divide the image into 4 equally sized blocks.
2) Find the best combination of scanning patterns out of 16 combinations.

The next section describes the tree structures explored and the basis for selection of a structure is stated.

B. Adaptive Grid Structure

The aim is to obtain a compact representation of the information content in each image and the content of an image can be described in terms of regions contained in the image. Thus, the amount of compression depends on pixel value statistics in a portion of the image [7][14]. The compression of 2D signal is achieved by segmenting the original signal into complex grid structure, which consequently results in various sources of pixel statistics with different compressibility. To partition an image, we use a tree-based search method to find an overall optimal tree structure for decomposing the image. The method involves decomposition of the image into variable size blocks using a tree structure. The original image is referred to as an *image tree* or *original tree*, which is then segmented. The image will be decomposed into larger blocks for smooth regions and smaller ones for regions with largely varying content. Adaptive grid structure takes advantage of correlations present due to uniformity in an image using image segmentation. It examines non-uniform areas and isolates them from the remaining areas of the image, which can be coded at a lower bit rate.

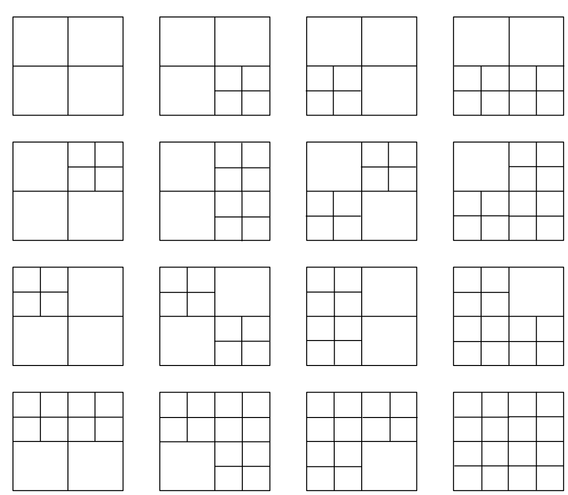


Fig. 3: 16 possible grid structure for partitioning an image.

As depicted in Fig. 3, a variable block size scheme is used to segment the image into complex grid structure. There are several ways to perform segmentation. In tree-based search algorithm, segmentation is performed with variable-size grid

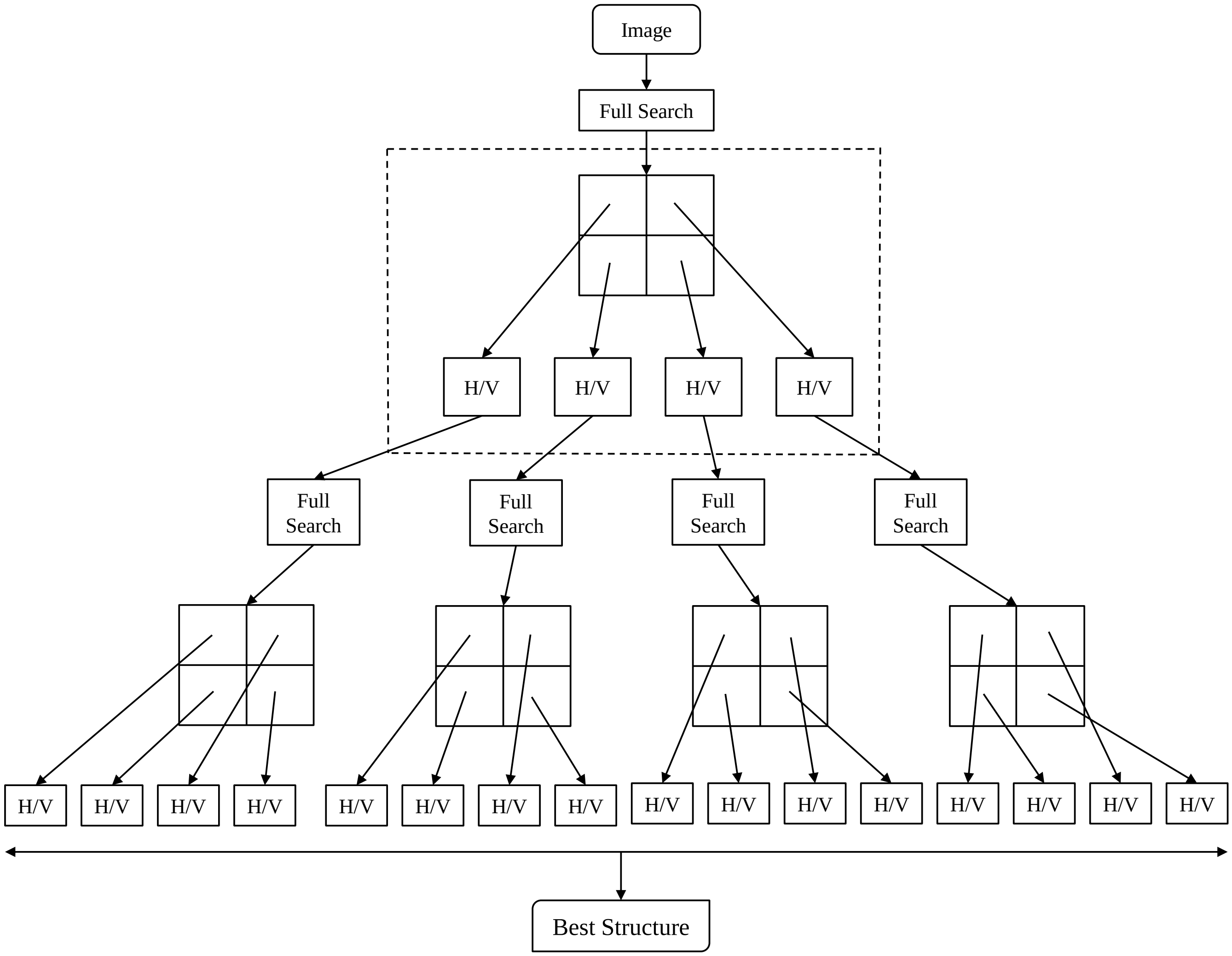


Fig. 4: Tree based search algorithm.

structure to divide non-uniform regions of the image into smaller blocks, from which we can identify larger "uniform" blocks. In performing such segmentation, the tree-based search algorithm takes a sequence of binary decisions based on the full search performed on the sub-blocks. A two-level tree structure has a total of 16 structures as shown in Fig. 3 and all of them are examined to find out the best among the whole set.

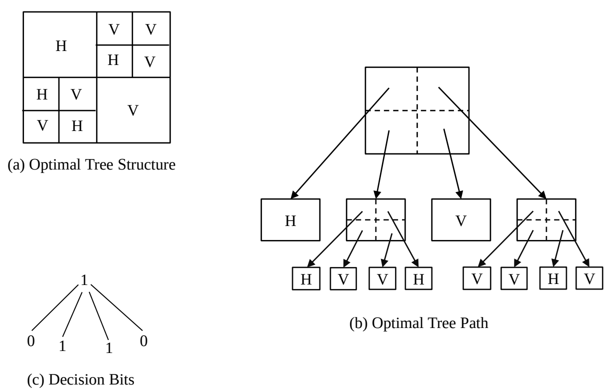
C. Two Level Recursive Splitting

Tree based search approach is a very effective and computationally simple technique for image compression. The workflow of the tree based algorithm is summarized by the flowchart shown in Fig. 4. The algorithm starts with decomposing the original image, also known as *original tree*, into two levels by partitioning the image recursively into four sub images. At level one, the original tree is divided into four blocks and at level two, the image blocks are further divided into four blocks. Therefore, as depicted in Fig. 4, an image can be represented by a tree structure. The segmentation is performed iteratively and checked at each step. Each of the 16 structures in Fig. 3 is examined to produce different interval sequences and the data file is compressed using bzip2 compressor to generate compression ratio for each structure to ensure that the segmented image improves compression. The tree structures are constructed by repeated splits of the image.

The proposed algorithm uses a tree structure in which the original image is the root node. The original image is split into four-sized blocks, emanating four branches from it. The root of the branch is referred to as parent node and each of the four branches are called child nodes. The child nodes are obtained after splitting the parent block. Each of the child node can further be subdivided based on the full search test [15]. The tree represents recursive splitting of the space (image), and a node at any stage represents one quarter sub-block of the image block from the previous stage. Therefore, every node of interest corresponds to one region in the original space. Four child nodes will occupy four different regions. If we put the four child nodes together, we get the same region as that of the parent node.

A series of bits, where bit ”0” indicates termination and bit ”1” indicates subdivision, is used to designate a tree structure. In the end, every leaf node is assigned with a class (‘H’ or ‘V’) based on the full search test which returns the efficient direction that should be used to traverse an image block [16]. Full search test is performed to determine whether segmentation of a block improves compression or not. If the test is positive, then the node becomes a terminal node with division, otherwise it becomes a leaf node and only one bit is required to indicate the information about the division. We refer to this bit as “binary tree structure bit” or “decision bit” since it represents the shape of the final tree selected to be encoded [17].

Let us consider the structure shown in Fig. 5(a) as the optimal tree structure. Fig. 5(b) shows the tree path followed and Fig. 5(c) corresponds to the division decision bits. The tree is defined in such a way that each node can either be a parent node indicating termination by a leaf node or can emanate four child nodes from it [18].



(a) Optimal Tree Structure

(b) Optimal Tree Path

(c) Decision Bits

Fig. 5: Final Structure.

A judgment of terminating or segmenting the block is made and if the block is partitioned, then a binary decision is made for selection of scanning direction. The process continues for the rest of the blocks. To identify the optimal tree, the structures shown in Fig. 3 are examined. Hence, the procedure terminates after two-level recursive splitting. The final non-uniform grid structure in tree-based algorithm is represented by the side information, which indicates if the division of sub-block is performed or not.

The direction bits and sequence of intervals are combined into a data file with the header, which consists of the image size and the tree structure. The final step is to compress the data file by a bzip2 compressor.

Lossless check is performed by comparing the original image with the decoded image pixel by pixel.

III. Simulation Results

This section discusses the simulation results of proposed algorithms and its comparison to six other lossless bi-level image compressors. MATLAB R2017a was used for code implementation and result analysis. To evaluate the performance of the proposed method, a set of binary images were generated using thresholding on the grayscale frames of the video sequence [19]. Fig. 6 shows 9 frames of the Table Tennis video sequence for demonstration.

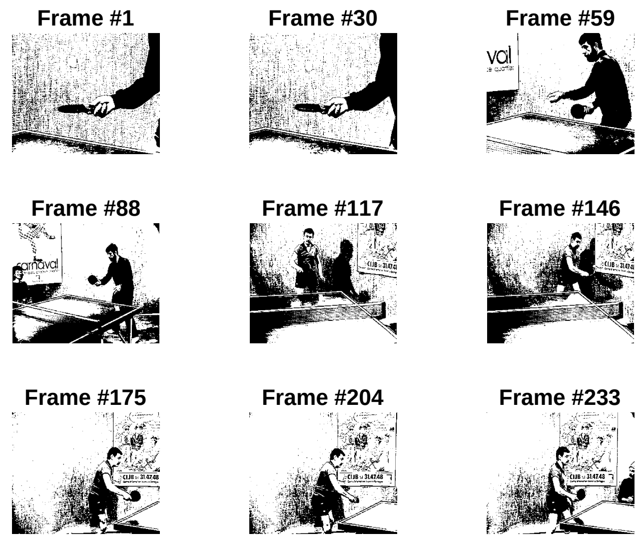


Fig. 6: Binary images (288×352) obtained from a video sequence.

A. Simulation Results

The binary images are fed to the proposed tree based algorithm. At the end, we generate a bitmap (“1”, i.e., white for horizontal and “0”, i.e., black for vertical scan) for viewing the scan path of the whole image. Typically, the bitmap gives a global view of the direction bits of the entire image.

Fig. 7 shows the bitmaps of the corresponding table tennis sequence frames shown in Fig. 6. Every image in a video sequence is either slightly or significantly different from one another. Therefore, similar frames give similar bitmaps, as demonstrated for frame number 204 and 233. Significantly different bitmaps are generated for significantly different frames. Different bitmaps implies that images can be scanned in a different way in order to achieve better compression.

B. Comparison with Other Techniques

In this section, we provide simulation results of the proposed algorithms. Fig. 8 shows the compression results for 9 frames in table tennis sequence. Test image index 1, 2, 3, 4, 5, 6, 7, 8 and 9 refers to frame 1, 30, 59, 88, 117, 146, 175, 204, and 233 in the sequence, respectively. The tree-based search algorithm was tested to examine its performance on binary images. Simulation results showed the scheme achieves the highest compression with significant difference for most of the images (frame 5 to frame 9) in the sequence.

The proposed methods were tested on hyperspectral binary maps. Similar improvements were obtained for different types

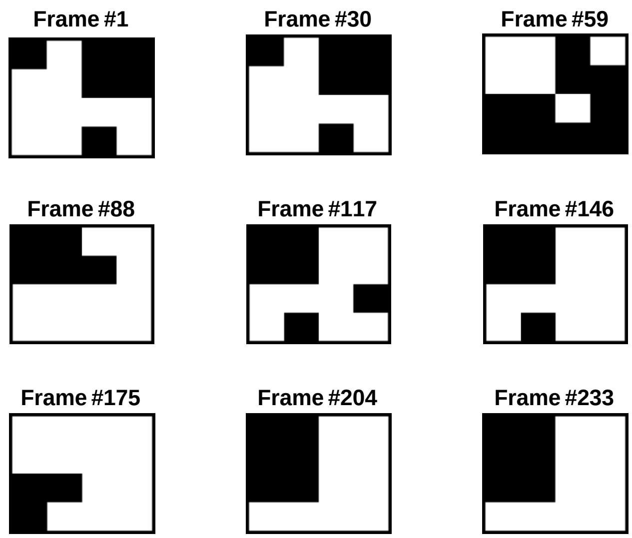


Fig. 7: Bitmaps of the "Tennis" sequence using tree-based search method.

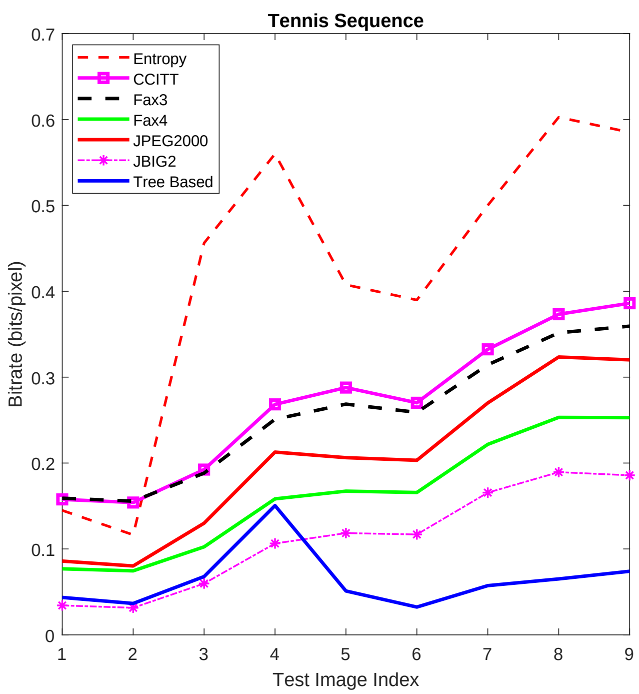


Fig. 8: Compression results for tennis sequence.

of images, demonstrating the advantage of dividing the image into variable size blocks.

Hyperspectral images: Engineers build hyperspectral sensors and processing systems for application in astronomy, agriculture, biomedical imaging, geosciences, physics, and surveillance [20][21][22]. A hyperspectral image is a three-dimensional image with two spatial dimensions and one spectral dimension. High compression can be achieved on hyperspectral dataset by storing region-of-interest (ROI's) [23]. Fig. 9 shows the ROI maps (600×340) of hyperspectral sample dataset ("Pavia University").

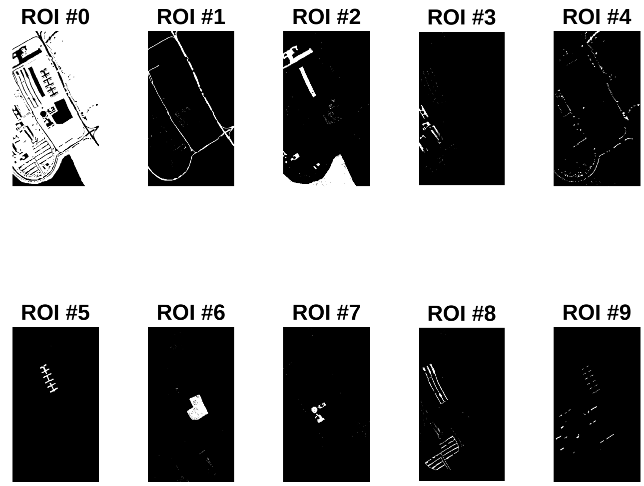


Fig. 9: Classification label maps of the "Pavia University (PU)" dataset.

Region of interests are samples within a data set identified for a purpose. The bi-level map describes the location of the ROI pixels. Machine learning algorithms such as Support Vector Machine (SVM) could be crafted to detect ROIs [24][25]. We tested the proposed algorithms on ROI maps shown in Fig. 9. We compressed all 9 ROI in the sample dataset (Pavia University) and tested five other compressors. The results are plotted in Fig. 10 for easy comparison.

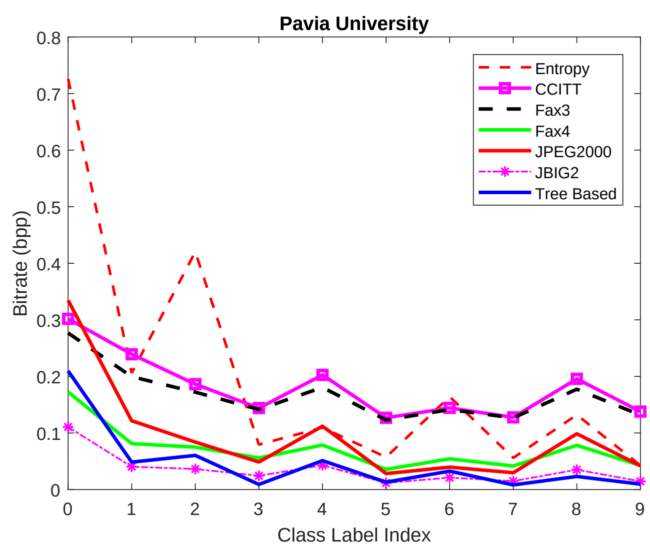


Fig. 10: Compression results for bi-level PU ROI maps.

We observe that for ROI maps 3, 7, 8, 9, the tree based algorithm outperforms all other compressors including JBIG2 standard and performs very closely to JBIG 2 standard method for rest of the ROI maps.

IV. Conclusion

In this work, we proposed a new compression scheme based on variable size blocks, which divides the image using an adaptive grid structure as determined by a tree-based search algorithm. The resulting non-uniform sized blocks can be scanned either horizontally or vertically in order to generate a one-dimensional data files to be compressed. The search is based on the criteria of minimal compressed file sizes. Simulation results demonstrate that the proposed coding method achieved highest compression on some binary images when compared to several other efficient lossless compression standards such as CCITT, FAX3, FAX4, JBIG2 and JPEG 2000. A general extension to this work could be to increase the number of scan paths to choose from (for instance, by using the Hilbert scan or zig-zag scan, etc.), which will result in generating different interval sequences. This will provide much larger search space to explore with the potential to achieve additional compression gains. In addition, the tree-based search scheme has only two levels, which limits the scope of the search. The two-level tree-based scheme can be extended to more levels, with the goal of increasing the compression significantly.

References

[1] A. L. Liaghati and W. D. Pan, "An adaptive interval generation method for efficient distance coding of binary images," in *Proc. SPIE*, vol. 9443, pp. 9443–9447, *Sixth Int. Conf. Graphic Image Process. (ICGIP 2014)* 2015.

[2] A. L. Liaghati and W. D. Pan, "Improved distance coding of binary images by run length coding of the most probable interval," in *IEEE SoutheastCon 2015, Fort Lauderdale, FL, USA*, pp. 1–7, Apr. 2015.

[3] H. Lin, J. Si, and G. P. Abousleman, "Region-of-interest detection and its application to image segmentation and compression," in *Int. Conf. Integr. Knowl. Intensive Multi-Agent Syst.*, pp. 306–311, Apr. 2007.

[4] P. Y. Lin, "Basic image compression algorithm and introduction to JPEG standard." Nat. Taiwan Univ., Taipei, Taiwan, ROC, 2009.

[5] G. E. Blelloch, "Introduction to data compression." Comput. Sci. Dept., Carnegie Mellon Univ., Jan 2013.

[6] S. Verma and A. K. Pandit, "Quad tree based image compression," in *TENCON 2008 - 2008 IEEE Region 10 Conf.*, pp. 1–4, Nov. 2008.

[7] R. Leonardi and M. Kunt, "Adaptive split-and-merge for image analysis and coding," *Proc. SPIE*, vol. 0594, pp. 0594 – 0594, 1986.

[8] A. Mulla, J. Baviskar, A. Baviskar, and C. Warty, "Image compression scheme based on zig-zag 3D-DCT and LDPC coding," in *2014 Int. Conf. Adv. Comput. , Commun. Inform. (ICACCI)*, pp. 2380–2384, Sep. 2014.

[9] W. A. Pearlman, "Trends of tree-based, set-partitioning compression techniques in still and moving image systems," in *In Still and Moving Image Systems," Proc. of Picture Coding Symp.*, pp. 1–8, Apr. 2001.

[10] W. Y. Wei, "An introduction to image compression." Nat. Taiwan Univ., Taipei, Taiwan, ROC, 2008.

[11] J. Z Wang, D. Geman, J. Luo, and R. Gray, "Real-world image annotation and retrieval: An introduction to the special section," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 1873–76, Nov. 2008.

[12] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Process.*, vol. 40, pp. 2040 – 2060, Aug. 1992.

[13] N. Memon, D. L. Neuhoff, and S. Shende, "An analysis of some common scanning techniques for lossless image coding," *IEEE Trans. Image Process.*, vol. 9, pp. 1837–1848, Nov. 2000.

[14] D. D. Ridder, *Adaptive Methods of Image Processing*. PhD thesis, Delft Univ. of Technol., Netherlands, Dec. 2001.

[15] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. on Image Process.*, vol. 3, pp. 2661–2664, May. 1991.

[16] W. Y. Loh, "Classification and regression trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, 2011.

[17] M. V. H. Radha and R. Leonardi, "Image compression using binary space partitioning trees," *IEEE Trans. on Image Process.*, vol. 5, pp. 1610 – 1624, Dec. 1996.

[18] A. Said and A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June. 1996.

[19] "Xiph.org video test media [derf's collection]." http://media.xiph.org/video/derf/.

[20] A. Liaghati, W. D. Pan, and Z. Jiang, "Biased run-length coding of bi-level classification label maps of hyperspectral," in *IEEE J. Sel. Topics Appl. Earth Observ. and Remote Sens. (JSTARS)*, vol. 10, pp. 4580–4588, Oct. 2017.

[21] H. Shen, W. D. Pan, and D. Wu, "Predictive lossless compression of regions of interest in hyperspectral images with no-data regions," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, pp. 173–182, Jan. 2017.

[22] Z. Jiang, W. D. Pan, and H. Shen, "Universal golomb–rice coding parameter estimation using deep belief networks for hyperspectral image compression," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, pp. 3830–3840, Oct. 2018.

[23] H. Shen, W. D. Pan, Y. Dong, and Z. Jiang, "Golomb-Rice coding parameter learning using deep belief network for hyperspectral image compression," in *IEEE Int. Geosci. Remote Sens. Symp., IGARSS'17, Fort Worth, TX, USA, July 23-28*, pp. 2239–2242, 2017.

[24] A. L. Liaghati, H. Shen, and W. D. Pan, "An efficient method for lossless compression of bi-level ROI maps of hyperspectral images," in *Proc. of IEEE Aerosp. Conf., Big Sky, Montana*, pp. 1–6, Mar. 2016.

[25] Z. Jiang, W. D. Pan, and H. Shen, "LSTM based adaptive filtering for reduced prediction errors of hyperspectral images," in *Proc. of IEEE 2018 Int. Conf. Wireless Space Extreme Environ. (WISEE'18), Huntsville, AL*, Dec. 2018.