Image Compression Using Binary Space Partitioning and Geometric Wavelets

Pranob K Charles[1], Dr.Habibulla Khan[2], Vinnakota Harish[3], Mule Swathi[3]
Chedurupalli Deepthi[3], Cherukuri Rajesh Kumar[3], Nikhita Nayudu[3]
*(Associate Professor, Dept. of Electronics and Communication Engineering, KLUniversity.)
** (Professor,HOD, Dept. of Electronics and Communication Engineering, KLUniversity.)
*** (B.Tech Scholars, Dept. of Electronics and Communication Engineering, KLUniversity.)

ABSTRACT

For low bit-rate compression applications, segmentation-based coding methods provide, in general, high compression ratios when compared with traditional coding approaches. During the last years, after JPEG2000, different techniques were developed in the area of Image Compression. Although they outperformed the JPEG2000 algorithm, the partitioning problem persists. In this paper, we present a segmentation based image compression technique which is based on Binary Space Partitioning (BSP) and Geometric wavelets. By using Binary Space partitioning technique the image is segmented recursively into a number of segments until an exit criterion is met and a tree is formed with all these segments. Geometric Wavelets are used to remove the insignificant nodes present in the tree. Finally, this method is compared with various wavelet based and transform based image compression techniques and it is show that this method outperforms all of them.

Keywords **– Binary Space Partitioning, image compression, geometric wavelets, JPEG2000.**

I. INTRODUCTION

The field of image compression is rich in diverse source coding schemes ranging from classical lossless techniques and popular transform approaches to the more recent segmentation-based coding methods. The notion of segmentation-based coding was introduced during the early 1980's.Segmentation-based compression methods usually describe the desired image as a set of regions.

In general, the description of each region requires two types of information: 1) the geometry of the region boundaries and 2) the attributes of the image signal within the region. In order to achieve high compression ratio and good image quality, one needs to segment the image into a minimum number of regions such that the geometric description of the regions' boundaries is simple and the image signal within each region is continuous (or smooth).
Therefore, the most challenging aspect of a segmentation-based coding approach is to balance between a small number of geometrically simple regions and the smoothness (or continuity) of the image signal within these regions.

The main work described in this review is based on the document [1] "an improved image compression algorithm using binary space partition scheme and geometric wavelets" written by G.Chopra, A.K.Pal and published in IEEE transactions on image processing in 2011, at some point it is discussed aspects of [3] "image compression using binary space partitioning trees" written by Hayder Radha, Martin Vetterli and Riccardo Leonardi, and published in IEEE transactions on image processing in 1996. and there is a support document to complete the discussion [2] "image coding with geometric wavelets" written by Dror Alani, Amir Averbuch and Shai Dekel and published in IEEE transactions on image processing in 2007.

The technique used in [1] is similar to [2] but they differ only in the case of selecting the type of partition line. The normal form of the straight line is used to represent the partition line incase of [2] whereas, the slope intercept form of the line is used in [1].

This method is applied to 8 bits gray scale images but it could be extended to color images in the same way that JPG2000 has been applied to different type of images (i.e. 8bits/pixel, 24bits/pixels).

In the following sections the algorithm, pseudo code, Binary Space Partition method, tree encoding, results and conclusions are described.

II. BINARY SPACE PARTITIONING (BSP)

Segmentation techniques partition the digital image into a set of different geometric regions which are approximated by simple functions. Segmentation based image coding methods were introduced during the early 1980[6], [7].Since then, many segmentation techniques have been developed and among them the BSP scheme is a simple and effective method.

The most challenging aspect of a segmentation based coding approach is to balance between a small number of geometrically simple regions and the smoothness of the image signal within these regions.

The BSP can be summarized as follows. Given an image *f*, the algorithm divides Ω into two subsets Ω_0 and Ω_1 using a bisecting line and minimizing a given functional. The algorithm continues partitioning each region recursively until it reaches a given measure or there is no enough pixels to subdivide. The algorithm constructs a binary tree with the partitioning information.

To approximate the image *f* in a given region Ω_i a bivariate linear polynomial is used which is defined by:

$$Q_{\Omega_i} = A_i x + B_i y + C_i \qquad (1)$$

The functional used to find the best subdivision for a given region is the following:

$$\mathrm{F}(\Omega_0, \Omega_1) = \arg\min_{\Omega_0,\Omega_1} \|f - Q_{\Omega_0}\|^2_{\Omega_0} + \|f - Q_{\Omega_1}\|^2_{\Omega_1}. \quad (2)$$

Where Ω_0 and Ω_1 represent the subsets resulting from the subdivision of Ω where Ω_0 and Ω_1 should be considered as children for the father Ω . Fig.1. shows the steps involved in Binary Space Partitioning algorithm. First a line L divides the region Ω into two regions Ω_0 and Ω_1 . The two regions Ω_0 and Ω_1 are further divided into Ω_{00} , Ω_{01} and Ω_{11} , Ω_{10} respectively. These four regions are further divided into eight segments and this is done recursively. Then it is represented in a tree structure as shown in fig.2.

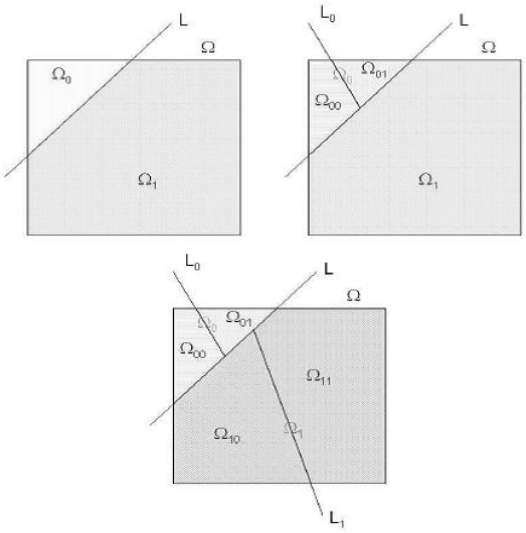

Fig.1. Two partition levels using bisecting lines

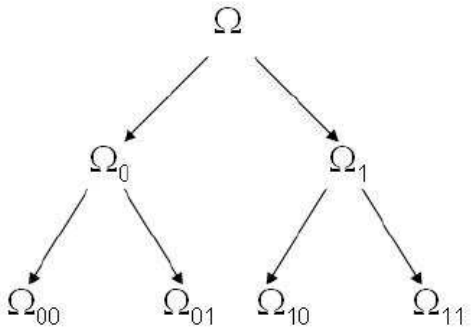

Fig.2. BSP tree representation

The value of the coefficients A,B and C are found by minimizing the function given below:

$$\prod = \sum_{i=1}^{n} [f(x_i, y_i) - (A\, x_i + B\, y_i + C)]^2 \qquad (3)$$

By taking the partial derivatives for A, B ,C and solving the below three equations , we will get the coefficients of the polynomial.

$$\sum_{i=1}^{n} z_i = A\, n + B \sum_{i=1}^{n} x_i + C \sum_{i=1}^{n} y_i$$

$$\sum_{i=1}^{n} x_i\, z_i = A \sum_{i=1}^{n} x_i + B \sum_{i=1}^{n} x_i^2 + C \sum_{i=1}^{n} x_i y_i$$

$$\sum_{i=1}^{n} y_i\, z_i = A \sum_{i=1}^{n} y_i + B \sum_{i=1}^{n} x_i y_i + C \sum_{i=1}^{n} y_i^2$$

III. SPARSE GEOMETRIC WAVELETS (GW) REPRESENTATION

In [1] they use the local difference to define the geometric wavelets. The local difference computes the difference between the actual partition and the previous giving us an idea of the degree of change, if the difference is large then the new partition is capturing new details, and if the difference is small, the new partition does not add new information. The GW is defined as follows:

$$\Psi\Omega_0(f) \triangleq 1\Omega_0(Q_{\Omega_0} - Q_\Omega) \qquad (4)$$

Where $1_{\Omega 0}$ is the function that gives us 1 in Ω_0 and 0 in the rest. Ω_0 here means one of the children.

We show that it is possible to reconstruct the function *f* using GW due to the term cancelations.

$$f = \sum_{\Omega_i} \Psi_{\Omega_i}(f) \qquad (5)$$

But using the BSP tree we can compute the norm of each $\psi_{\Omega i}$ (*f*), which is a measure of the degree of change, then sorting these numbers it is possible to approximate the function by the n-term geometric wavelet sum defined as

$$f \approx \sum_{j=0}^{n} \Psi_{\Omega k_j}(f) \qquad (6)$$

The BSP tree that is generated may contain a large number of GW nodes. Yet, for low bit-rate coding, only few of them (typically 1-4 %) are needed to obtain a reasonable approximation of the image. Therefore, we apply the 'greedy' approximation methodology; sort the geometric wavelets according to their energy 'contribution' and extract a global n-term approximation from the joint list of all the geometric wavelets over all the image tiles.

We found that for the purpose of efficient encoding it is useful to impose the additional condition of a tree structure over each image tile. Namely, we require that if a child appears in the sparse representation, then so does the parent. Once a parent is encoded in this hierarchical representation, then we only need to encode the (quantized) BSP line that creates the child. This significantly saves bits when the geometry of the sparse representation is encoded. On the other hand, the penalty for imposing the connected tree structure is not significant, since with high probability, if a child is significant, then so are his ancestors. Fig. 3 illustrates an n-term GW collection whose graph representation includes some unconnected components and Fig. 4 illustrates the final GW tree after the missing ancestors were added.

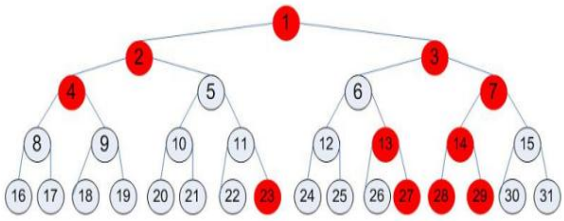

Fig.3. Example of a greedy selection.

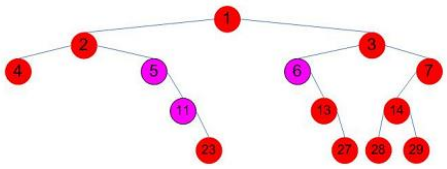

Fig.4. The final GW tree with the additional nodes.

Finally, we apply a rate-distortion (R - D) optimization process. Empirical results show that this rate-distortion mechanism increases the PSNR by 0.1 dB in some cases.

IV. ENCODING THE SPARSE GW REPRESENTATION

Once the BSP tree is generated the next step is to discard the irrelevant information by computing $\psi_{\Omega ki}$ (f) .

Before the actual BSP Tree is encoded, a small header is written to the compressed file. This header contains the minimum and maximum values of the coefficients of the wavelets Q_Ω participating in the sparse representation. These values are used by the decoder to decode the coefficients. In addition, the header contains the minimum and maximum values of the gray levels in the image. The coefficients extremal values are encoded with four bytes each and image extremal values with 1 byte each. Therefore, the header size is $3 \times 2 \times 4+2 \times 1= 26$ bytes.

Due to the fact that the leaves are necessary for the reconstruction, in [3] they impose the requirement that if a child appears in the tree then father has to appear too. But it is not necessary that both children appear in the tree, if one is not significant enough, could be excluded from the sparse representation, then all its descendant should be excluded too. This is an improvement with respect to [2] because if a partition is done both the children appear in the tree independent of whether one child is significant or

not.After the tree is pruned it is encoded using the following information:

- Tree structure information.
    - Number of children.
    - Information to distinguish each child node.
- The quantized coefficients Q_{Ω}.
- The bisecting line information of each Ω if it has a child.
- Header information.

The tree-structure is encoded using the fact that with a high probability a significant node does not have a significant child, in a similar way like 'zero-trees'. Therefore using Huffman code to encode the three different values (Zero children='1', One Child='01', Two-Children='00') it is possible to save in some cases 1 bit, due to normally it is necessary 2 bit to encode 3 different states.

The quantized coefficient Q - that represent the wavelet polynomials are determined by three real numbers (A_i ; B_i ; C_i) that can be stored with 12 bytes using the standard 4-bytes float representation. But in [3] they show an algorithm to store at a rate of 1.5 bytes per polynomial on average, using the standard Graham-Schmidt method to obtain the orthonormal base representation. This could be the greatest improvement with respect to [2].

In order to deal with the time consuming algorithm , we tile the image in squares of 128x128 and they apply the BSP algorithm on each tile. The main disadvantage of doing this is that blocking artifacts appears at the tiles' boundaries. Another disadvantage is that connected areas could be disconnected missing the possibility to improve the approximation.Fig.5 shows the tiling of the cameraman image.

Fig.5. Tiling of cameraman image of size 256 x 256

V. DECODING

In this step compressed bit stream is read to find whether the participating node is the leaf node, has 1 child or 2 children. If one child is participating then by using bit stream, it is found that whether it is left or right. If at least one of the children belongs to the sparse representation, then the coefficients of the bisecting line are calculated. Thereafter, using this optimal cut, domain is partitioned into two sub-domains; and depending upon the situation vertex set of only one child or both children is found. An orthogonal basis was used during the encoding of the coefficients of the coefficients of geometric wavelet. Thus, before using the decoded geometric wavelets in n-term sum, its representation in the standard basis is found. This process is repeated until entire bit stream is read.

VI. ALGORITHM

A. Binary Space Partitioning.

1. Read the Image.
2. Tile the image using tiles of size 128 X 128.
3. Select a title.
4. Select a partition line using least square error based criteria.
5. Divide the title into two segments.
6. Repeat steps 4 and 5 until a minimum threshold value is reached for a particular segment.
7. Repeat steps 4,5 and 6 for the next tile.
8. Taking tile as a parent and the corresponding segments as child we form a tree with the leaf nodes as the final segments.

B. Sparse Geometric Wavelet Representation

9. We apply greedy approximation method for the tree obtained in 'A'.
10. If any child has a missing parent we include it.

C. Encoding:

11. A header containing the minimum and maximum values of the coefficients of the wavelets as well as the

minimum and maximum of the grey levels of the image is added to the compressed files.
12. Encoding the tree structure.

13. Encoding the bisecting line.

14. Encoding the coefficients of the wavelet polynomials.

15. Quantizing the coefficients in an orthogonal polynomial basis representation.

16. Bit allocation of polynomial coefficients.

D. Decoding:

17. Compressed bit stream is read.
19. Whether the selected node is leaf node, has 1 child or 2 children is found.

20. If one child is present it is found whether it is left or right.

21. The coefficients of the polynomial and the bisecting line are decoded.

22. This process is repeated until the entire bit stream is read.

VII.PSEUDO CODE

```
I = image_read('image_name');
        //reads the image to I

I_new = image_tile(I);

        //tiles the image to I_new

For i=1:1:4

    I = select_tile(I_new);

    While (threshold is not reached)

        Select_partition_line();

        Divide_image();

    End

End

Tree = Construct_Binary_Tree( );

        //creates the BSP tree

GD_tree = Greedy_approximate( Tree );

        //applies greedy approximation

I_encode = Encode_tree( GD_tree );

        //encodes the tree structure

I_decode = Decode( I_encode );

        //reconstructs the image
```

VIII. RESULTS

The PSNR (peak signal to noise ratio) based on MSE (mean square error) is used as a measure of "quality." MSE and PSNR are given by the following relations:

$$MSE = \frac{1}{m * n} \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{i,j} - y_{i,j})^2$$

$$PSNR = 10 \log[(255)^2 / MSE]$$

Where n x m is the image size, $x_{i,j}$ is the initial image $y_{i,j}$ is the reconstructed image. MSE and PSNR are inversely proportional to each other and higher value of the PSNR produces better image compression.

Fig 6. Original Image[1]

Fig. 7. Reconstructed image using the proposed method[1]

Table 1. Comparing the PSNR values of various methods[1]

Method	128:1	64:1	32:1
SPIHT	22.8	25	**28**
Kakadu	21.15	24.11	27.29
GW	22.93	25.07	27.48
Proposed method	**23.04**	**25.29**	27.62

IX. CONCLUSIONS

The key idea behind this work is the fact that it is possible to subdivide a region using a bisecting line and to encode this line only with a few bits depending on the quantization schema. This new approach to encode images seems to be better to encode images at a very low bit-rate.

The algorithm was exhaustively analyzed to reduce to the minimum the number of bits encoded, we can see the effort of the authors in 2.3.1 in [2], in order to reduce at most one bit in some cases, to encode the tree-structure.

Although it seems to be a good technique there exists a few things that are not clearly specified neither in [1] nor in [2].The authors in [1] state that the algorithm is computationally intensive. They do not show how much intensive it is. Due to the fact that in a brute force algorithm like this, it is not easy to show the order, but they should compare the performance of this algorithm in such a way some-one can estimate the encoding and decoding time, like running the algorithm in different computers or comparing times with the standard JPG2000.

One important idea, ones can infer from this is that a complex partition can give a better approximation but it means that more information is needed to store the partition, therefore a better storing algorithm is needed.

REFERENCES

[1] Garima Chopra and A.K.Pal , "An Improved Image Compression Algorithm Using Binary Space Partition Scheme and Geometric Wavelets",*IEEE Transactions on Image Processing.VOL 20,NO.1,January 2011.*

[2] D. Alani, A. Averbuch and S. Dekel, "Image coding with geometric wavelets", *IEEE Transactions on Image Processing, 16(1), 69-77, 2007.*

[3] H. Radha, M. Vetterli and R. Leonardi "Image Com-pression Using Binary Space Partitioning Trees", *IEEE Transactions on Image Processing, vol. 5,num. 12, pp. 1610-1624, 1996.*

[4] R. Shukla, L. Dragotti, M.N. D and M. Vetterli, "Rate-Distortion Optimized Tree-Structured *Compression Al-gorithms for Piecewise Polynomial Images", IEEE Transactions on Image Processing, vol. 14,num. 3, pp. 343-359, 2005.*

[5] M.Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients", *IEEE Transac-tions on Signal Processing, vol. 41, pp.3445-3462,1993.*

[6] M. Kocher and M.Kunt , "A contour-texture approach to image coding,", *in Proc. ICASSP, 1982, pp.436-440*

[7] M.Kunt, A.Lkonomopoulos, and M.Koche," Second generation image coding techniques", Proc. IEEE ,vol.73, no.4,pp,549-574, Apr.1985.

[8] M. A. Losada, G. Tohumoglu, D. Fraile, and A. Artes, "Multi-iteration wavelet zerotree coding for image compression," *Sci. Signal Process., vol. 80, pp. 1281–1287, 2000.*

[9] G.K.Wallace,"The JPEG still-picture compression standard," *Commun. ACM, vol. 34, pp. 30–44, Apr. 1991.*

[10] MPEG-2video,ITU-T-Recommendation H.262-ISO/IEC 13818-2, Jan. 1995.

[11] K. R. Rao and P. Yip, Discrete Cosine Transform:*Algorithms,Advantages,Applications. New York: Academic, 1990.*

[12] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process., vol. 41, no. 12, pp. 3445–3462, Dec. 1993.*

[13] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set portioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 243–250, Jun. 1996.*

[14] A. Islam and W. A. Pearlman, "An embedded and efficient low complexity ierarchical image coder," *in Proc. SPIE, Jan. 1999, vol. 3653, pp. 294–305.*

[15] D. Tauban, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process., vol. 9, no. 7, pp. 1158–1170, Jul. 2000.*