Asian Journal of Scientific Research

ISSN 1992-1454

Asian Journal of Scientific Research 7 (4): 418-433, 2014
ISSN 1992-1454 / DOI: 10.3923/ajsr.2014.418.433

Effect of Tiling on the Performance of GW Algorithm for Image Coding

[1]V.J. Rehna and [2]M.K. Jeya Kumar

[1]Department of Electronics and Communication Engineering, University of Noorul Islam, Kumarakoil, Tamil Nadu, India

[2]Department of Computer Application, University of Noorul Islam, Tamil Nadu, India

Corresponding Author: V.J. Rehna, Department of Electronics and Communication Engineering, Noorul Islam Center for Higher Education, Noorul Islam University, Kumarakoil, Tamil Nadu, India

ABSTRACT

As our dependence on the digital media continues to grow, finding competent ways of storing and conveying these large amounts of data has become a major concern. The technique of image compression has then become very essential and highly applicable. In this regard, the performance of an efficient segmentation-based image coding method combined with Geometric Wavelets that divides the desired image using a recursive procedure for image coding is investigated. The objective of the work is to optimize the performance of geometric wavelet based image coding scheme and to suggest a method to reduce the time complexity of the algorithm. We have used the polar coordinate form of the straight line in the BSP scheme for partitioning the image domain. A novel pruning algorithm is tried to optimize the rate distortion curve and achieve the desired bit rate. The algorithm is also implemented with the concept of no tiling and its effect in PSNR and computation time is explored. The enhanced results show a gain of 2.24 dB over the EZW algorithm and1.4 dB over the SPIHT algorithm at the bit-rate 0.0625 bpp for the Lena test image. Image tiling is found to reduce considerably the computational complexity and in turn the time complexity of the algorithm without affecting its coding efficiency. The algorithm provides remarkable results in terms of PSNR compared to existing techniques.

Key words: Segmentation, binary space partition, geometric wavelets, image coding, time complexity

INTRODUCTION

Digital images have become an integral part of our daily life, be it storage or transmission. As our reliance on the digital media continues to grow, finding proficient ways of storing and conveying these large amounts of data has become a major concern. Because the amount of space required to hold unadulterated images can be extremely large in terms of cost, as well as of the huge bandwidth required to transmit them, researchers are seeking methods for efficient representations of these digital pictures (Netravali and Haskell, 1988) to simplify their transmission and save disk space. At this point in time, the technique of image compression has become very essential and highly applicable. 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 (or second generation) coding methods (Froment and Mallat, 1992). The Discrete Cosine Transform (DCT) (Rao and Yip, 1990) has been, until recently, the most popular technique for image compression because of its optimal performance and ability to be implemented at a reasonable cost. The popular JPEG standard (Wallace, 1991) for still images and the MPEG standard for moving images are based on DCT. Wavelet-based image coding techniques are the latest development in the field of image compression offering multiresolution capability resulting in superior energy compaction and high quality reconstructed images at low bit rates. The discrete wavelet transform forms the basis of the popular JPEG 2000 (Skodras *et al.*, 2001). The wavelet transforms (Antonini *et al.*, 1992) based coding approaches have taken over other classical methods particularly the cosine transform, due to its capability to solve the problem of blocking artifacts which is a common phenomenon in DCT based compression. However, the EZW, the SPIHT, the SPECK, the EBCOT (Islam and Pearlman, 1999) algorithms and the current JPEG 2000 standard are based on the discrete wavelet transform (DWT) (Daubechies, 1990).

Despite providing outstanding results in terms of rate-distortion compression, the transform-based coding methods do not take an advantage of the geometry of the edge singularities in an image. This led to the design of 'Second Generation' or the segmentation based image coding techniques (Kunt *et al.*, 1985) that make use of the underlying geometry of edge singularities of an image. To this day, almost all of the proposed 'Second Generation' algorithms are not competitive with state of the art (dyadic) wavelet coding (Daubechies, 1992). In this regard, inspired by a recent progress in multivariate piecewise polynomial approximation, we put together the advantages of the classical method of coding using wavelets and the segmentation based coding schemes to what can be described as a geometric wavelet approach.

This study focuses on a recent development in the field of piecewise polynomial approximation for image coding using Geometric wavelets (Alani *et al.*, 2007). This scheme efficiently captures curve singularities and provides a sparse representation of the image and thereby achieves better quality reconstructed images with higher compression ratios. Stress is given on the shared approach of image compression using geometric wavelets and the binary space partition scheme. The algorithm is extremely complex in computation and has very high execution time. In general, the BSP algorithm is applied on individual tiled regions of the image. Once the BSP tree is generated from on tile the procedure is repeated on other tiles thereby creating a BSP forest. In this work, we have applied the BSP tree generation procedure on the whole image rather than on the individual tiled regions and compared the execution times. The effect of no tiling on PSNR is also explored.

LITERATURE REVIEW

A number of segmentation algorithms have been proposed for image coding till date, each claiming to be different or superior in some way. The first segmentation-based coding methods were suggested in the early 1980s. These algorithms partition the image into complex geometric regions using a contour-texture coding method (Kocher and Kunt, 1982) over which it is approximated using low-order polynomials. One of the most popular segmentation based coding schemes investigated by researchers in the early days were the Quadtree-based image compression (Sullivan and Baker, 1991) which recursively divides the image signal into simpler geometric regions. Many variations of the 'Second Generation' coding schemes have since been announced that exploit the geometry of curve singularities of an image (Kunt *et al.*, 1987). Coding algorithms that are geometric enhancements of existing wavelet transform based methods (Cohen and Matei, 2001), where wavelet coefficients are coded using geometric context modelling also exist. But all of these constructions are redundant, i.e., the output of the discrete transform implementations produces more coefficients than the original input data. Research on the possibility of using these new transforms to outperform wavelet based coding is still on-going.

The Binary Space Partition (BSP) scheme (Dekel and Leviatan, 2005), a simple and efficient method for hidden-surface removal and solid modelling was introduced in 1990. The BSP technique (Radha *et al.*, 1996) was applied to the concept of image compression in 1996 and is adopted in the first stage of this study. Later, in 2000, binary partition trees were used as an efficient representation for image processing, segmentation and information retrieval (Salembier and Garrido, 2000). Recently, many second generation image compression algorithms such as the Bandelets (Le Pennec and Mallat, 2005), the Prune Tree (Shukla *et al.*, 2005), the GW image coding method (Alani *et al.*, 2007) and the like based on the sparse geometric representation have been introduced. Le Pennec and Mallat (2005) lately applied their 'Bandelets' algorithm to image coding, where a warped-wavelet transform is computed to align with the geometric flow in the image and the edge singularities are coded using one-dimensional wavelet type approximations. The concept of combining the binary space partition scheme and geometric wavelets for compression of digital images were put forward by Alani *et al.* (2007). Here the bisecting lines of the BSP scheme are quantized using the normal form of straight line. This method successfully competes with state-of-the-art wavelet methods such as the EZW (Shapiro, 1993), SPIHT (Said and Pearlman, 1996) and EBCOT (Taubman, 2000) algorithms and also beats the recent segmentation based methods. But the algorithm turned out to be computationally intensive. An improvement was made to this work by Chopra and Pal (2011). They used the slope intercept form of a straight line instead of the normal representation. This improved the possibility of minimizing the cost functional by increasing the choice of bisecting lines available for partitioning. This technique further increased the complexity of the algorithm.

The presented approach deviates from the context of multi-scale geometric processing, even from the more general framework of harmonic analysis which is the theoretical basis for transform based methods and also from the popular wavelet based studies and is based on the GW and binary space partition method introduced in 2007 (Alani *et al.*, 2007). The main difference between the GW algorithm and recent work is that we use the polar coordinate representation of straight line for partitioning the domain thereby further improving the availability of partitioning lines and intern further minimizing the cost functional at each step of BSP scheme. Moreover, unlike the existing method of tiling the image into smaller blocks and applying the BSP algorithm independently on each tile, we adopted the method of no tiling, i.e., we apply the algorithm on the whole image. Image tiling has many disadvantages. The tiling procedure may significantly reduce the time complexity of the algorithm but also reduces its coding efficiency. Also at low bit-rates, there are blocking artifacts at the tiles' boundaries, similar to the common phenomenon in low bit-rate JPEG compression. In addition, there is a possibility that a long, linear portion of a curve singularity will be captured by several BSPs, one at each tile, whereas, with no tiling, only a single BSP is needed. Here, the BSP scheme is applied on the entire image by using the polar coordinate form of the straight line. We indicate the performance of the algorithm in terms of PSNR and execution times with and without tiling and show that image tiling significantly reduces the time complexity of the algorithm without affecting its coding efficiency.

THE CONCEPT

The basic concepts of the geometric wavelet method are described in the following sections:

BSP technique: The BSP technique can be described as follows. Given an image f, the algorithm divides convex polygonal domain Ω into two subsets Ω_0 and Ω_1 using a bisecting line. The subdivision is performed to minimize a given cost functional Eq. 1. This partitioning process then operates recursively in a hierarchical manner on the subdomains until some exit condition is met. To be specific, we describe the algorithm of which is a BSP algorithm that identifies a compact geometric description of a target bivariate function. The goal in is to encode an optimal cutoff the BSP tree, to be precise, a sparse piecewise polynomial approximation of the original image based on the union of disjoint polygonal domains in the BSP tree. Rate-distortion optimization strategies are used to meet a given bit rate.

For a given convex polygonal domain Ω, the algorithm finds two subdomains, Ω_0 and Ω_1 two bivariate (linear) polynomials $Q_{\Omega 0}$ and $Q_{\Omega 1}$, that minimizes the given cost functional:

$$F(\Omega_0, \Omega_1) = \arg\min_{\Omega_0 \Omega_1} \left\| f - Q_{\Omega 0} \right\|_{\Omega_0}^2 + \left\| f - Q_{\Omega_1} \right\|_{\Omega_1}^2 \tag{1}$$

where Ω_0 and Ω_1 represent the subsets resulting from the subdivision of Ω (Ω_0 and Ω_1 should be considered as children for the mother Ω). The bivariate polynomial used is defined by:

$$Q_{\Omega_i} = A_i x + B_i y + C_i \tag{2}$$

The polynomial interpolation is made using the least square method, computing the difference between the image and the polynomial at a defined region Ω. The algorithm continues partitioning each region recursively until there are no enough pixels to subdivide or the approximation error is sufficiently small. The algorithm constructs a binary tree with the partitioning information (Paterson and Yao, 1990). The algorithm needs to encode the information of the geometry, namely, the line that cut each sub-domain and the approximation function in each sub-domain represented by the polynomial coefficients. Figure 1 show 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 and so on until area of the subdomain contains only a very few pixels. Then it is represented in a tree structure as shown in Fig. 2.

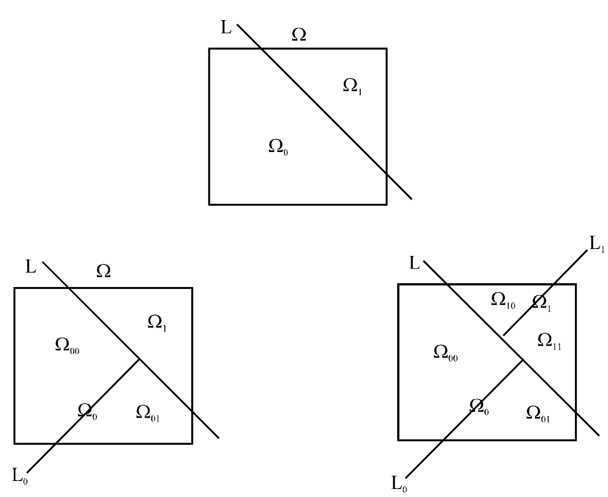

Fig. 1: Binary space partitioning of the domain Ω (two levels)

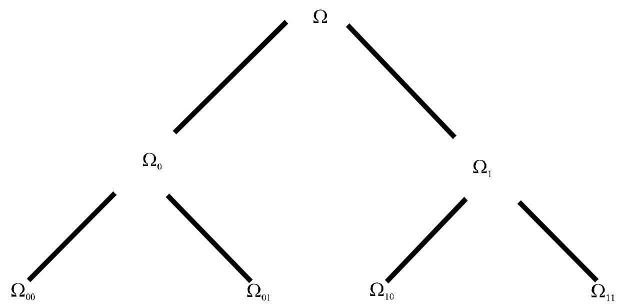

Fig. 2: BSP tree representation of the polygon in Fig. 1

Geometric wavelet: Geometric wavelets are multi-scale dictionary elements which are constructed directly from the data and have guarantees on the computational cost, the number of elements in the dictionary and the sparsity of the representation. Geometric Wavelets (GW) have been considered in context of image compression in 2007. It is a new multi-scale data representation technique which is useful for a variety of applications such as data compression, interpretation and anomaly detection. The GW is defined as:

$$\Psi_{\Omega 0}(f) \triangleq 1_{\Omega 0}(Q_{\Omega 0}-Q_{\Omega}) \tag{3}$$

Ω_0 here means one of the children of mother, Ω. It is possible to reconstruct the function fusing:

$$f = \Sigma_{\Omega i}\Psi_{\Omega i}(f) \tag{4}$$

Geometric Wavelet, Ψ_{Ω} is a "local difference" component that belongs to the detail space between two levels in the BSP tree, a "low resolution" level associated with Ω and a "high resolution" level associated with Ω_0. Geometric wavelets also satisfy the vanishing moment property like isotropic wavelets, i.e., if f is locally a polynomial over Ω, then minimizing of (1) gives $Q_{\Omega 0} = Q_{\Omega} = 1$ and therefore $\Psi_{\Omega 0}(f) = 0$. Unlike classical wavelets, geometric wavelets do not satisfy the two scale relation and the biorthogonality property.

THE GEOMETRIC WAVELET ENCODING ALGORITHM

As in wavelet decomposition, we only encode the differences between the original coarse projections of the data and the points projected onto the planes at a finer scale, to find a compact representation for the data at the finer scale. In order to do this, an effective scheme is developed based on the construction of a minimal space spanning this set of differences. The axes of this difference space are termed "geometric wavelets" and the projections of the finer-scale corrections to the data points onto the plane spanned by these axes are called the "wavelet coefficients". The process is continued, forming a binary tree of mother and children at finer and finer scales until no further details are needed to approximate the data up to a pre-specified accuracy. The process is discussed in detail in the following sections.

BSP tree construction: The BSP method is computationally very intensive. Therefore generally, the image is tiled first and then the BSP algorithm is applied independently on each tile

(Chopra and Pal, 2011). The tile size is generally adopted is 128×128. But image tiling has many disadvantages. The tiling procedure significantly reduces the time complexity of the algorithm but also reduces its coding efficiency. Also at low bit-rates, there are blocking artifacts at the tiles' boundaries. Several BSPs, one at each tile are to be created which is computationally very intensive, whereas, with no tiling, only a single BSP is needed. The BSP scheme is applied on the entire image by using the polar coordinate form of the straight line (Rehna and Jeyakumar, 2014). In polar coordinates on the euclidean plane, a line is expressed as:

$$r\text{-}\frac{b}{\sin\theta - m\cos\theta} \tag{5}$$

where m is the slope of the line and b is the y-intercept. The equation can be rewritten as:

$$r\sin\theta = m\cos\theta + b \tag{6}$$

It is not possible to quantize the parameter m, as it is unbounded, has value infinity for the straight lines which are parallel to y axis. This problem is solved by using the new parameter ø in place of m in (6), where ø is the angle between the line and the x axis in the anticlockwise direction (Fig. 3).

Subsequently, Eq. 6 reduces to:

$$r\sin\theta = \tan\Phi . r\cos\theta + b \tag{7}$$

Here, the probability of minimizing the cost functional given in (1) is increased, compared to that when the normal form of straight line is used. The number of bisecting lines available for the partitioning of the Cameraman test image of dimension 256×256 is 182325. Figure 4 shows the tiling of bitmap image of Cameraman of size 256×256 with each tile being 128×128.

We use a quantization scheme to discretize the set of possible bisecting lines by the parameters, r and θ of the polar co-ordinate form. The size of the quantization step depends on the "size" of the polygonal domain. The "size" is defined as the longest side of the domain's bounding box, but we find this too crude and use the diameter of the bounding box instead. We apply a uniform quantization of the line orientations θ, where the quantization depends on the size of the domain. We then use uniform quantization for, where the quantization depends on θ. For the purpose of quantization, we set the domain's size to be the smallest power-of-two that is larger than the diameter of the bounding box. Then, the number of line-orientations is:

$$\#\theta \triangleq \min_{2^j} \geq -\sqrt{mn}^{2^j} \tag{8}$$

where m and n are the lengths of the sides of the bounding box. Next, we quantize r. As mentioned before, given a domain Ω, the range of r is a function of θ. Let be the corner (x_c, y_c) of the bounding-box of Ω. Then, the range where r takes values is computed by:

$$r_{\min}(\theta) \triangleq \min_{(x,y\in V)}\{(x - x_c)\cos\theta + (y - y_c)\sin\theta\} \tag{9}$$

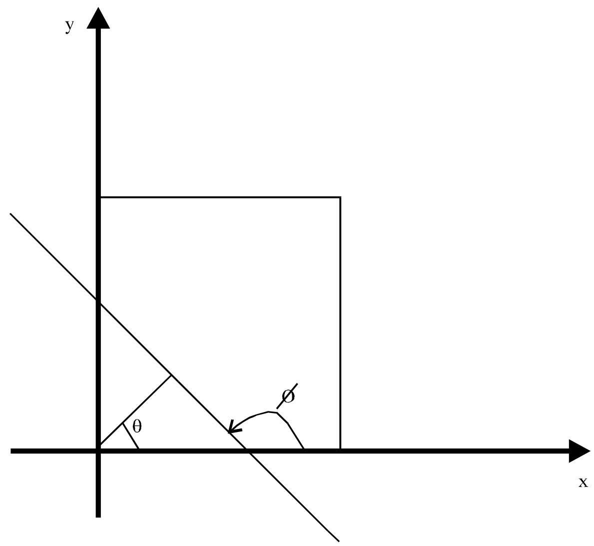

Fig. 3: Partition of the image domain into two subdomains-Parameters θ and Φ

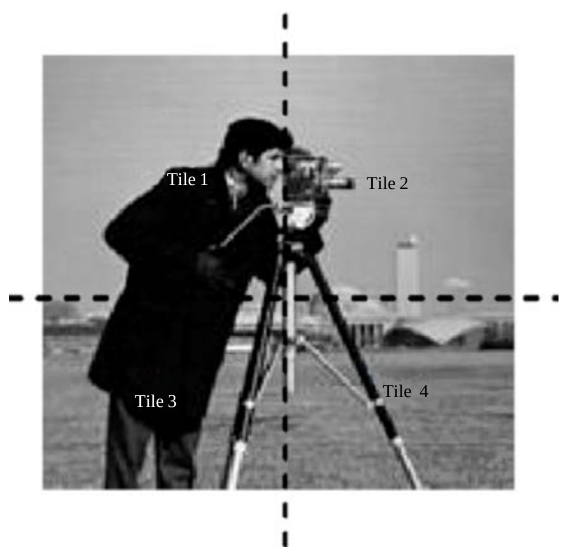

Fig. 4: Tiling of cameraman image (Tile size 128×128)

$$r_{mix}(\theta) \triangleq mix_{(x,y\in V)}\{(x - x_c)\cos\theta + (y - y_c)\sin\theta\} \tag{10}$$

where V is the set of vertices of the polygonal boundary of Ω and r is the fixed line orientation. Given a line-orientation θ, the quantization step of r is:

$$\Delta_r(\theta) = \max\{|\cos\theta|, |\sin\theta|\} \tag{11}$$

Equation 11 gives the smallest value for a quantization step size that reveals a new set of pixels with each step. Following the procedure mentioned in earlier sections, the BSP tree is generated for the entire image and according to the method defined in earlier section; geometric wavelets are created for each node.

Pruning and sparse representation: The GW image coding algorithm is based on the idea that among all the geometric wavelets only a "few" wavelets have large norm. Once all the geometric wavelets are created, they are arranged according to their L^2 norm as shown in Eq.12:

$$\left\|\Psi_{\Omega_{k_1}}\right\|_2 \geq \left\|\Psi_{\Omega_{k_2}}\right\|_2 \geq \left\|\Psi_{\Omega_{k_3}}\right\|_2 \geq \cdots \tag{12}$$

Then the sparse geometric representation is extracted using the greedy methodology of nonlinear approximation (Claypoole *et al*., 2003). Here, n wavelets are selected from the joint list of geometric wavelets over the entire image.

We tried a new rate-distortion optimization pruning algorithm prior to encoding. The R-D curve for each node is generated by approximating the node by the quantized polynomial $\hat{p}(t)$ which is obtained by scalar quantizing the polynomial coefficients. In this Lagrangian cost based pruning, the R-D optimal pruning criterion for the given operating slope, λ is as follows: Prune the children if the sum of Lagrangian costs of the children is greater than or equal to the Lagrangian cost of the parent. Mathematically, this means that the children are pruned if $(D_{C1}+D_{C2}) + \lambda(R_{C1}+R_{C2}) >= (D_P+\lambda R_P)$, where R_P and D_P are rate and distortions of the parent and R_{C1}, R_{C2}, D_{C1} and D_{C2} are the rate and distortions of the children respectively. Subsequently, function f is approximated using the n-term geometric wavelet sum given in Eq. 4 where n is the number of wavelets used in the sparse representation.

Encoding: To obtain a reasonable approximation of the image, it is essential that if a child is present in the sparse representation, then the mother should also be there, i.e, the BSP tree should be connected. Therefore, instead of encoding an n-term tree approximation, we create an n + k geometric wavelet tree by considering more k nodes. The cost of imposing the condition of the connected tree structure is not very huge, since there is high probability that if a child is important all its ancestors are also important.

The encoding of the geometry of the extracted connected tree structure saves bits as only optimal cut is to be encoded. There are two sorts of data to be encoded, (1) the geometry of the support of the wavelets participating in the sparse representation and (2) the polynomial coefficients of the wavelet. Before encoding the extracted BSP forest, a small header is written to the compressed file. Header consists of the minimum and maximum values of the coefficients of the participating wavelet and the image gray levels. Out of header size of 26 bytes, 24 are used in the storage of the minimum and the maximum values of the coefficients while 2 bytes are utilized to store the extremal values of the image. "Root" geometric wavelets contribute most in the approximation, so each root wavelet is encoded. Thus the encoding process is performed over the entire image.

Encoding the geometry of the support of the wavelet: The following information is encoded for each of the participating node Ω:

- Number of children of Ω that participate in the sparse representation
- In case only one child is participating, then whether it is the left or the right child
- If Ω is not a leaf node, then the line that bisects Ω is encoded using the slope intercept form

Left child and right child are defined as the sets of the pixels satisfying the inequality r-tan ø. r sinθ<= b and r-tan ø. r sinθ>= b, respectively. The leaf node is encoded by using the bit "1." Codes "00" and "01" are used for the one child symbol and the two children symbol, respectively. If only 1 child of Ω is participating in the sparse representation, then this event is encoded by using an additional bit. In case Ω is not a leaf node, then the indices of the parameter ø and c of the bisecting line are encoded using the lossless variable length coding.

Encoding the wavelet coefficients: The coefficients of the wavelet polynomial, Q_Ω are quantized and encoded using an orthonormal representation of $\Pi_1(\Omega)$, where $\Pi_1(\Omega)$ is the set of all bivariate linear polynomials over Ω. A bit allocation scheme for the coefficients is applied using their distribution function (over all the domains) which is discussed in later sections. The "root" wavelet is always encoded.

Quantizing the wavelet coefficients: To ensure the stability of the quantization process of the geometric wavelet polynomial Q_Ω, we first need to find its representation in appropriate orthonormal basis. The orthonormal basis of $\Pi_1(\Omega)$ is found using the standard Graham-Schmidt procedure. Let $V_1(x,y) = 1$, $V_2(x,y) = x$, $V_3(x,y) = y$ and be the standard polynomial basis. Then, an orthonormal basis of $\Pi_1(\Omega)$ is given by:

$$U_1 = \frac{V_1}{\|V_1\|}$$

$$U_2 = \frac{V_2 - \langle V_2, U_1 \rangle U_1}{\|V_2 - \langle V_2, U_1 \rangle U_1\|}$$

$$U_3 = \frac{V_3 - \langle V_3, U_1 \rangle U_1 - \langle V_3, U_2 \rangle U_2}{\|V_3 - \langle V_3, U_1 \rangle U_1 - \langle V_3, U_2 \rangle U_2\|} \tag{13}$$

where inner product and norm are associated with the space L_2 (Ω). Let:

$$\Psi = aU_1 + \beta U_2 + \gamma U_3 \tag{14}$$

be the representation of the geometric wavelet $\Psi \epsilon \Pi_1(\Omega)$ in the orthonormal basis.

A bit allocation scheme is applied depending upon the distribution functions of the coefficients α, β and ã of the wavelets participating in the sparse representation. Figure 5 shows the histogram of the wavelet coefficients of Lena. We can infer from the graph that there is a very high probability for the coefficients α, β and γ to be small (the graph resembles a generalized-Gaussian function). Some large coefficients are also present due to root wavelets. Four bins are used to model the absolute value of the coefficients; bin limits are computed and passed to the decoder. In case all the three coefficients of the wavelet are small, i.e., they are present in the bin containing zero, then

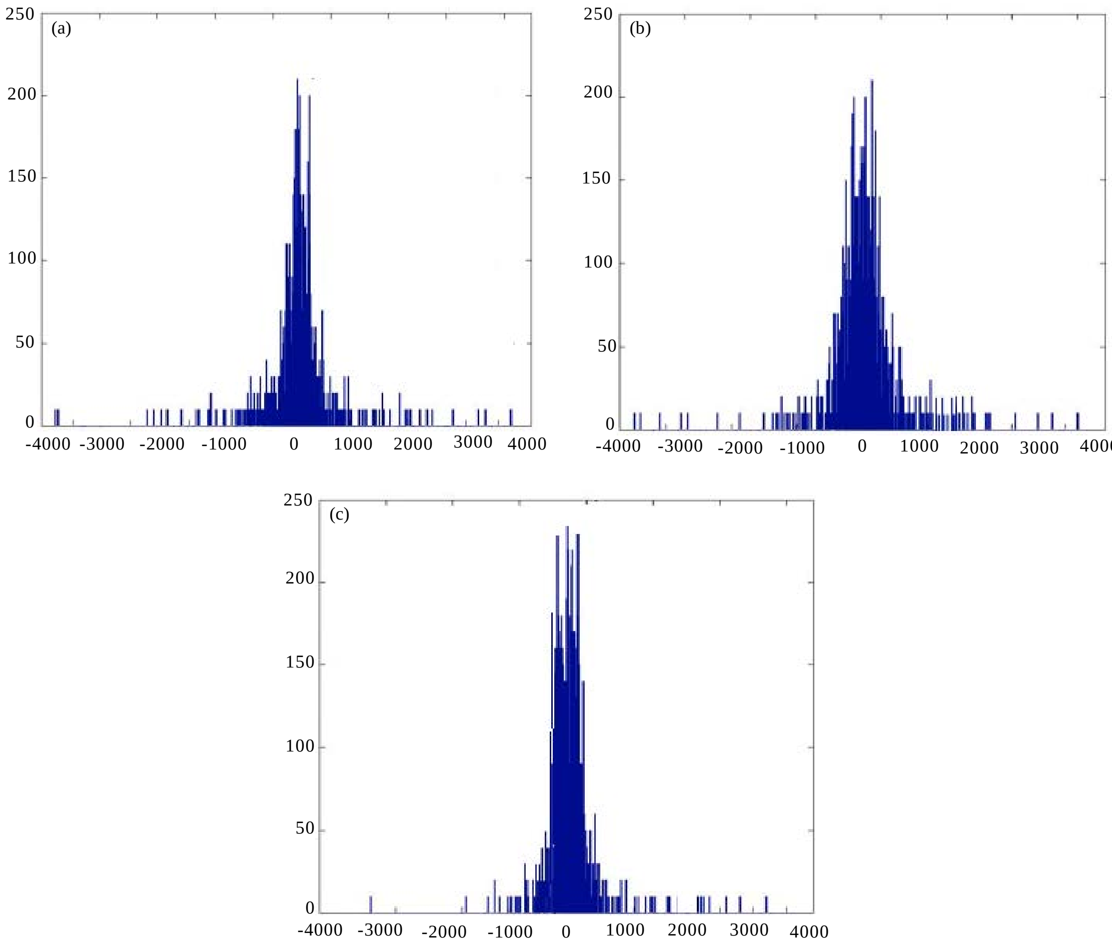

Fig. 5(a-c): Histogram of wavelet coefficients, (a) α (b), β and (c) γ of Lena image

this event is encoded using single bit, but if any one of them is not small then the bin number of each coefficient is encoded. After this quantized bits are written to the compressed file.

Figure 6 shows how the bit budget allocation of Lena at the bit-rate 0.03125 bits per pixel (bpp) is distributed among the GW algorithm components. Figure 7 shows the bit allocation distribution of Lena at the bit-rate 0.125 bits per pixel (bpp). It can be inferred from the chart that at higher bit-rates, the bit budget for the polynomial coefficients relatively increases, while the bit allocation for the bisecting lines decreases.

Decoding: In the decoding stage, the compressed bit stream is read to find whether the participating node is a root node, has 1 child or 2 children, or a leaf node. If one child is participating then by using bit stream identification, it is found whether it is left child or right child. If at least one of the children belongs to the sparse representation, then the indexes of ø and b are decoded and using these index parameters ø and b of optimal cut are calculated. Thereafter, using

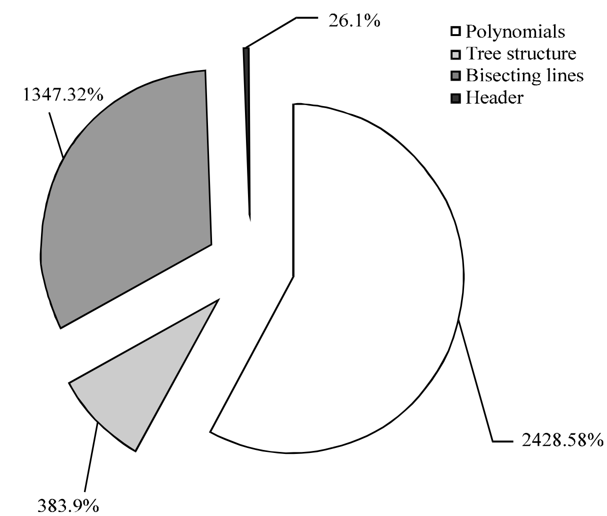

Fig. 6: Bit budget allocation for Lena at bit-rate 0.03125 bpp. Output file size is 1 kByte

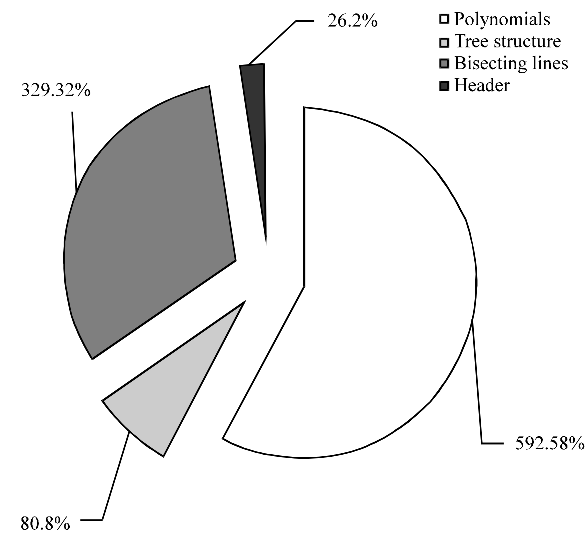

Fig. 7: Bit budget allocation for Lena at bit-rate 0.125 bpp. Output file size is 4 kBytes

this optimal cut, domain is partitioned into two subdomains and depending upon the situation vertex set of only one child or both children is found. This process is repeated until entire bit stream is read.

EXPERIMENTAL RESULTS

The proposed algorithm is tested on the still image of Lena of bit depth 8 and of size 512×512. The implementation is done using MATLAB. The Peak Signal to Noise Ratio (PSNR) based on Mean Square Error (MSE) is used as a measure of "quality". MSE and PSNR are given by the following relations:

$$MSE = \frac{1}{mxn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{i,j}-y_{i,j}\right)^2 \qquad (15)$$

$$\text{PSNR} = 10\log_{10}\left[\frac{(255)^2}{\text{MSE}}\right] \tag{16}$$

where m×n is the image size, $x_{i,j}$ is the original image and $y_{i,j}$ is the reconstructed image. MSE and PSNR are inversely proportional to each other and higher value of the PSNR implies better quality reconstructed image. The performance of proposed method with and without tiling is compared against six algorithms. The PSNR values obtained by this method for the Lena image are compared with those obtained by the EZW, the SPIHT, the EBCOT and the Bandelets algorithms. Data presented in Table 1 shows that the proposed method outperforms the EZW, the SPIHT, the EBCOT and the Bandelets methods (Le Pennec and Mallat, 2005) at low and medium bit rates.

The proposed method without tiling reports a gain of 1.4 dB over the SPIHT method, 1.48 dB over the EBCOT method and 2.24 dB over EZW algorithm at the compression ratio of 128:1 for the Lena test image. The presented algorithm shows an increase of 1.06 dB over the original GW method and 1 dB over the improved GW algorithm at a bit rate of 0.0625 bpp for the Lena image.

The PSNR comparison with other algorithms on the cameraman test image is shown in Table 2. At the compression ratios of 128, 64 and 32, proposed method performs better than the SPIHT, GW and the improved GW algorithms. The proposed method with tiling reports a gain of 1.24 dB and without tiling shows a gain of 1.07 dB over the GW method at the compression ratio of 64 for the cameraman image.

Figure 8 shows the graphical representation of the effect of tiling on PSNR at different bit-rates for the Lena and cameraman test image.

Different variations of the Geometric gorithm namely, the basic GW (Alani *et al.*, 2007) which uses the normal form of straight line for partitioning the image domain in the BSP scheme, Improved GW (Chopra and Pal, 2011) that uses the slope-intercept representation of straight line in the BSP scheme, hybrid GW (Rehna and Jeyakumar, 2013a) that uses the polar co-ordinate form of straight line and the proposed method are compared in Fig. 9.

Table 3 shows the execution times for BSP tree generation for all four tiles of the cameraman image. The simulated results presented are for the computer specifications of Intel core i5 processor, 3GB DDR3 memory and a speed of 1.64 GHz (Rehna and Jeyakumar, 2013b).

Table 1: Comparison of PSNR in dB with other state of the art algorithms on test image, Lena

								Proposed method	
Compression ratio	Bit rate (bpp)	EZW	SPIHT	EBCOT	Bandlets	GW	Improved GW	Tiling	No tiling
256:1	0.03125	25.38	26.1	-	-	26.64	26.67	27.84	27.90
128:1	0.0625	27.54	28.38	28.30	-	28.72	28.78	29.73	29.78
64:1	0.125	30.23	31.10	31.22	30.63	30.73	30.82	31.45	31.49

Table 2: Comparison of PSNR in dB with other state of the art algorithms on cameraman test image

					Proposed method	
Compression ratio	Bit rate (bpp)	SPIHT	GW	Improved GW	Tiling	No tiling
128:1	0.0625	22.8	22.93	23.04	24.55	24.63
64:1	0.125	25	25.07	25.29	26.31	26.36
32:1	0.25	28	27.48	27.62	28.38	28.42

Table 3: Time complexity analysis of BSP tree generation for different tiles of the cameraman image

		Execution time	
Tile No.	Tile size	Sec	Min
1	128×128	4325.26	72.08
2	128×128	5960.23	99.37
3	128×128	2632.43	43.87
4	128×128	3460.47	57.66

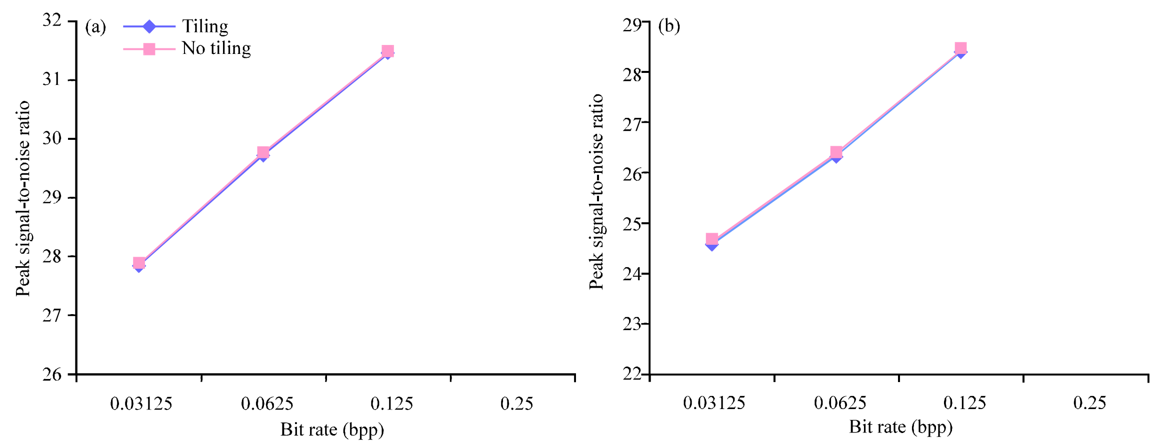

Fig. 8(a-b): Effect of tiling on PSNR for (a) Lena and (b) Cameraman image

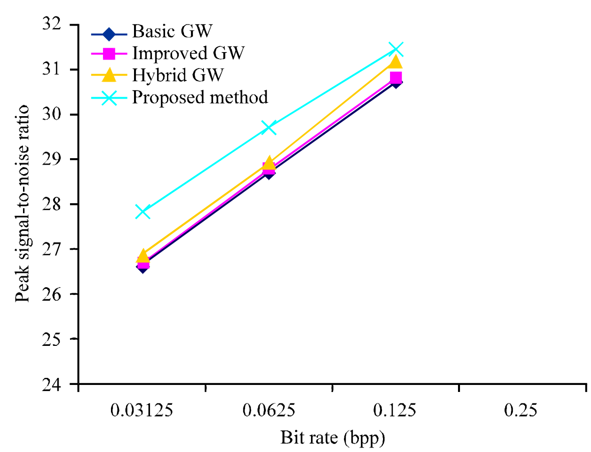

Fig. 9: Performance comparison for different variations of the GW algorithm

The total time required to generate BSP tree for the image with tiling is about 4.54 h. When the algorithm is applied to the entire image, i.e., without tiling, the execution time is found to be around 11.36 h. It can be seen from Table 3 that the peak signal to noise is very minimally affected without tiling. In other words, there is no considerable reduction in the coding efficiency due to tiling. It can be inferred from the results that image tiling significantly reduces the time complexity of the algorithm without reducing its coding efficiency. Figure 10 shows the reconstructed images of Lena using the GW algorithm with and proposed method without tiling, at the compression ratio of 128:1 and PSNR 28.72 and 29.78, respectively.

Fig. 10(a-f): (a) Original Lena. (512×512), (b) Reconstructed Lena using the GW algorithm (with tiling), 0.0625 bpp, PSNR = 28.72, (c) Reconstructed Lena using the proposed method (without tiling), 0.0625 bpp, PSNR = 29.78, (d) Original cameraman. (256×256). (e) Reconstructed cameraman using the GW algorithm (with tiling), 0.0625 bpp, PSNR = 22.93, (f) Reconstructed cameraman using the proposed method (without tiling), 0.0625 bpp, PSNR = 24.63

SUMMARY AND CONCLUDING REMARKS

The performance of a hybrid algorithm for image compression using the geometric wavelets and the tree-structured binary space partition scheme is explored in this research. We have improved the coding efficiency of the GW algorithm by using the polar coordinate form of straight line for best bisection in the partitioning procedure. A novel pruning algorithm is tried to optimize the rate distortion curve and achieve the desired bit rate. A new “geometric” context modeling scheme combined with arithmetic coding is designed to boost the performance of the algorithm. The algorithm is extremely complex in computation and has very high execution time. The time complexity of the BSP scheme is analyzed in this work. In addition, the algorithm is implemented with the concept of no tiling and its effect in PSNR and computation time is explored. Image tiling is found to reduce considerably the computational complexity and in turn the time complexity of the algorithm without affecting its coding efficiency. The presented method produced PSNR values that are competitive with the state-of-art coders in literature. The algorithm works well with geometrically rich content images at low bit-rates. Our results provide some weak evidence that show, hybrid techniques do help in improving the performance of image coding algorithms. We suggest future researchers to apply the concepts of soft computing techniques, especially the artificial neural networks with the geometric wavelets which may help in reducing the time complexity of the algorithm.

REFERENCES

Alani, D., A. Averbuch and S. Dekel, 2007. Image coding with geometric wavelets. IEEE Trans. Image Process., 16: 69-77.

Antonini, M., M. Barlaud, P. Mathieu and I. Daubechies, 1992. Image coding using wavelet transform. IEEE Trans. Image Process., 1: 205-220.

Chopra, G. and A.K. Pal, 2011. An improved image compression algorithm using binary space partition scheme and geometric wavelets. IEEE Trans. Image Process., 20: 270-275.

Claypoole, R.L., G.M. Davis, W. Sweldens and R.G. Baraniuk, 2003. Nonlinear wavelet transforms for image coding via lifting. IEEE Trans. Image Process., 12: 1449-1459.

Cohen, A. and B. Matei, 2001. Compact representation of images by edge adapted multiscale transforms. Proceedings of the International Conference on Image Processing, Volume 1, October 7-10, 2001, Thessaloniki, Greece, pp: 8-11.

Daubechies, I., 1990. The wavelet transform, time-frequency localization and signal analysis. IEEE Trans. Inform. Theory, 36: 961-1005.

Daubechies, I., 1992. Ten Lectures on Wavelets. SIAM, Philadelphia, PA., USA., ISBN: 9780898712742, Pages: 357.

Dekel, S. and D. Leviatan, 2005. Adaptive multivariate approximation using binary space partitions and geometric wavelets. SIAM J. Numer. Anal., 43: 707-732.

Froment, J. and S. Mallat, 1992. Wavelets: A Tutorial in Theory and Applications. In: Second Generation Compact Image Coding with Wavelets, Chui, C.K. (Ed.) Academic Press, New York, USA.

Islam, A. and W.A. Pearlman, 1999. An embedded and efficient low complexity hierarchical image coder. Proc. SPIE, 3653: 294-305.

Kocher, M. and M. Kunt, 1982. A contour-texture approach to picture coding. Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 7, May 3-5, 1982, Paris, France, pp: 436-439.

Kunt, M., A. Ikonomopoulos and M. Kocher, 1985. Second-generation image coding techniques. Proc. IEEE, 73: 549-574.

Kunt, M., M. Benard and R. Leonardi, 1987. Recent results in high-compression image coding. IEEE Trans. Circuits Syst., 34: 1306-1336.

Le Pennec, E. and S.G. Mallat, 2005. Sparse geometric image representations with bandelets. IEEE Trans. Image Process., 14: 423-438.

Netravali, A.N. and B.G. Haskell, 1988. Digital Pictures: Representation and Compressions. Plenum Press, New York, ISBN: 9780306427916, Pages: 586.

Paterson, M.S. and F.F. Yao, 1990. Efficient binary space partitions for hidden-surface removal and solid modeling. Discrete Comput. Geom., 5: 485-503.

Radha, H., M. Vetterli and R. Leonardi, 1996. Image compression using binary space partitioning trees. IEEE Trans. Image Process., 5: 1610-1624.

Rao, K.R. and P. Yip, 1990. Discrete Cosine Transform: Algorithms, Advantages and Applications. Academic Press, New York, USA., ISBN-13: 9780125802031, Pages: 490.

Rehna, V.J. and M.K. Jeyakumar, 2013a. A superior hybrid algorithm based on geometric wavelets for compression of digital images. Pensee J., 75: 304-314.

Rehna, V.J. and M.K. Jeyakumar, 2013b. An efficient hybrid algorithm for low bit rate image coding. Proceedings of the International Conference on Artificial Intelligence in Computer Science and ICT, November 25-26, 2013, Langkawi, Malaysia, pp: 21-28.

Rehna, V.J. and M.K. Jeyakumar, 2014. An enhanced geometric wavelet based hybrid image compression algorithm for low bit rate applications. Eur. J. Sci. Res., 116: 544-555.

Said, A. and W.A. Pearlman, 1996. A new, fast and efficient image codec based on set partitioning in hierarchical trees. IEEE Trans. Circuits Syst. Video Technol., 6: 243-250.

Salembier, P. and L. Garrido, 2000. Binary partition tree as an efficient representation for image processing, segmentation and information retrieval. IEEE Trans. Image Process., 9: 561-576.

Shapiro, J.M., 1993. Embedded image coding using zerotrees of wavelet coefficients signal processing. IEEE Trans. Signal Process., 41: 3445-3462.

Shukla, R., P.L. Dragotti, M.N. Do and M. Vetterli, 2005. Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. IEEE Trans. Image Process., 14: 343-359.

Skodras, A., C. Christopoulos and T. Ebrahimi, 2001. The JPEG 2000 still image compression standard. IEEE Signal Process. Mag., 18: 36-58.

Sullivan, G.J. and R. Baker, 1991. Efficient quadtree coding of images and video. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, April 14-17, 1991, Toronto, Canada, pp: 2661-2664.

Taubman, D., 2000. High performance scalable image compression with EBCOT. IEEE Trans. Image Process., 9: 1158-1170.

Wallace, G.K., 1991. The JPEG still picture compression standard. Commun. ACM, 34: 30-44.