A HYBRID ALGORITHM USING GEOMETRIC WAVELETS FOR LOW BIT RATE IMAGE CODING

Rehna. V. J¹, Dr. M. K. Jeya Kumar²
Noorul Islam University, INDIA
¹rehna_vj@yahoo.co.in
²jeyakumarmk@yahoo.com

Abstract
Image compression systems based on hybrid image coding techniques that combine the advantages of various classical methods have been developed over the years. Segmentation-based coding methods provide high compression ratios compared to traditional coding approaches such as transform and subband coding, especially for low bit-rate compression applications. In this paper, a hybrid algorithm using segmentation based binary space partition scheme and geometric wavelets, well suited for low bit rate image coding is discussed. The present study improves the geometric wavelet image compression method by using the slope intercept depiction of the straight line in the binary space partition scheme. The scheme takes advantage of the underlying geometry of the edge singularities in an image. This method is compared with other state-of-the-art wavelet based techniques as well as other recent segmentation based methods and shown that this method outperforms all of them at low bit rates. The results show a gain of 0.79 dB over the EZW algorithm at a bit rate of 0.03125.

Keywords: image coding, segmentation, binary space partition, edge singularity, bit rate

1. Introduction
These days, the digital world is more focused on storage space and speed. With the growing demand for improved bandwidth utilization, efficient image data compression techniques have emerged as an important factor for image data transmission and storage. The main goal of image compression is to reduce irrelevance and redundancy of the image data thereby optimizing the storage space and increasing the transmission rate over WebPages. Image compression is performed in such a way that it enables image reconstruction. The amount of compression achieved depends on the contents of image data. A typical photographic image can be compressed to about 80% of its original size without experiencing noticeable degradation in the quality. The strong relationship between image data items enables reduction in data contents without noteworthy quality degradation.

To date, different approaches to image compression have been developed like the classical predictive coding, popular transform coding and vector quantization. Several second generation coding schemes [1] or the segmentation based schemes are also gaining popularity. The compression ratio, starting at 1 with the first digital picture in the early 1960s, has reached a saturation level around 300:1 a couple of years ago. Image quality still remains as an important problem to be investigated. As the compression ratio increases, the quality of the resulting image degrades. So, a tradeoff between compression ratio and the tolerance in the visual quality degradation need to be considered during compression.

This paper focuses on a hybrid image coding scheme [3] which combines the advantages of the classical method of coding using wavelets and a segmentation based method, so as to efficiently capture curve singularities and provide a sparse representation of the image and thereby achieve better quality reconstructed images with higher compression ratios. Stress is given on the combined approach of image compression using geometric wavelets and binary space partition scheme [1].

Rest of the paper is organized as follows: Section II deals with the basic concepts of segmentation based and wavelet based image compression schemes, section III gives the details of the geometric wavelet image coding algorithm, section IV provides experimental results and presents a comparison of this method with recent state-of-the-art wavelet and “sparse geometric representation” methods. Discussion & conclusion of the work is presented in section V.
2. Basic Concepts

Segmentation based compression methods, in general, describe the desired image as a set of regions. In most cases, the description of each image region requires two types of information, namely, the geometry of the region boundaries and the attributes of the image signal within the region [1]. 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 region’s boundaries is simple and the image signal within each region is continuous (or smooth). The most difficult aspect of a segmentation-based coding approach is to balance between the small number of geometrically simple regions and the smoothness (or continuity) of the image signal within these regions [2].

2.1. Binary Space Partitioning

Segmentation techniques partition the digital image into a set of different geometric regions which are approximated by simple functions. Since 1980, many segmentation techniques have been developed and among them the BSP (Binary Space Partitioning) scheme [1] is a simple and effective method. The key idea behind this technique is to recursively perform subdivisions over the image domain and to approximate these regions by a function, typically a low order polynomial.

The BSP scheme can be summarized as follows. Given an image \( f \), the algorithm divides \( \Omega \) into two subsets \( \Omega_0 \) and \( \Omega_1 \) using a bisecting line that minimizes the given functional [2]:

\[
F(\Omega_0, \Omega_1) = \arg \min_{l_{01}} \| f - Q_{\Omega_0} \|_{\Omega_0}^2 + \| f - Q_{\Omega_1} \|_{\Omega_1}^2
\]

(1)

Where \( \Omega_0 \) and \( \Omega_1 \) represent the subsets resulting from the subdivision of \( \Omega \) where \( \Omega_0 \) and \( \Omega_1 \) should be considered as children for the father \( \Omega \). 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. Figure 1 shows the steps involved in Binary Space Partitioning algorithm [2].

![Figure 1: Partitioning using bisecting lines (Two levels)](image)

First a line \( L \) divides the region \( \Omega \) into two regions \( \Omega_0 \) and \( \Omega_1 \). The two regions \( \Omega_0 \) and \( \Omega_1 \) are further divided into \( \Omega_{00}, \Omega_{01}, \Omega_{10}, \Omega_{11} \) 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 Figure 2.

![Figure 2: BSP Tree Representation](image)
The algorithm needs to encode the information of the geometry, that is, the line that cut each sub-domain, and the approximation function in each sub-domain, that is represented by the polynomial coefficients. The polynomial interpolation is made using the least square method [5], computing the difference between the image and the polynomial at a defined region Ω.

### 2.2 Geometric Wavelet

Geometric wavelets (GW) have been considered in context of image compression in [12]. 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 [9]. The GW is defined as:

\[
Ψ_{Ω0}(f) \triangleq 1_{Ω0}(Q_{Ω0} - Q_{Ω})
\]  

(2)

where 1Ω0 is the function that gives us 1 in Ω0 and 0 in the rest. Ω0 here means one of the children. It is possible to reconstruct the function f using:

\[
f = \sum_{Ωi} Ψ_{Ωi}(f)
\]  

(3)

Like isotropic wavelets [4], ΨΩ 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 Ω0. Geometric wavelets also satisfy the vanishing moment property i.e., if f is locally a polynomial over Ω then minimizing of (1) gives QΩ0 = QΩ = 1, and therefore ΨΩ0 (f) = 0. However, unlike classical wavelets, geometric wavelets do not satisfy the two scale relation and the biorthogonality property.

### 3. The GW Algorithm

To form a compact representation for the data at a finer scale, 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 the finer scale [10]. In order to do this an efficient scheme is derived based on the construction of a minimal space spanning this set of differences. The axes of this difference space are called “geometric wavelets”, and the projections of the finer-scale corrections [11] 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 parents and children at finer and finer scales until no further details are needed to approximate the data up to a pre-specified precision. The process is discussed in detail in the following sections.

#### 3.1 BSP Forest Construction

The BSP method is computationally very intensive. Therefore, the image is tiled first and then the BSP algorithm is applied separately on each tile, thereby creating a BSP forest. The tile size is generally taken as 128 x 128. The number of bisecting lines available for the partitioning of tile of dimension 128 x 128 in [6] is 15740. But in the proposed algorithm this availability number increases to 60775. Hence, this method has better possibility to minimize the cost functional (1). The BSP scheme is applied on each tile of the image by using the slope intercept form of the straight line, i.e.,

\[
y = mx + c
\]  

(4)

where \(m\) is the slope and \(c\) is the intercept on the y axis. 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 as in (5).

\[
\rho = x \cos \theta + y \sin \theta
\]  

(5)

where \(\rho\) is the normal distance between the line and an origin point close to the subdivided domain and \(\theta\) is the angle between the line’s normal and axis.
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 [7]. This problem is solved by using the new parameter $\phi$ in place of $m$ in (8), where $\phi$ is the angle between the line and the $x$ axis in the anticlockwise direction (Figure 3).

![Figure 3: Parameters $\phi$ and $c$ of slope intercept form of line](image)

Subsequently, equation (4) reduces to:

$$ y = \tan \phi \cdot x + c $$  

(5)

### 3.2 Sparse Representation

The GW image coding algorithm [12] is based on the fact that among all the geometric wavelets only a “few” wavelets have large norm. Once all the geometric wavelets are created, they are sorted according to their $L^2$ norm, i.e,

$$ \left\| \Psi_{\Omega_{k1}} \right\|_2 \geq \left\| \Psi_{\Omega_{k2}} \right\|_2 \geq \left\| \Psi_{\Omega_{k3}} \right\|_2 \geq \cdots $$  

(6)

Then the sparse geometric representation is extracted using the greedy methodology of nonlinear approximation. Here, $n$ wavelets are selected from the joint list of geometric wavelets over all tiles. Thereafter, function $f$ is approximated using the $n$-term geometric wavelet sum given in (3), where $n$ is the number of wavelets used in the sparse representation [8].

### 3.3 Encoding

For efficient encoding of extracted BSP forest, it is necessary that if a child is present in the sparse representation, then the parent should also be there, i.e, the BSP tree should be connected [12]. Instead of encoding an $n$-term tree approximation, we create an $n + k$ geometric wavelet tree by considering more $k$ nodes. The penalty for imposing the condition of the connected tree structure is not very massive, since there is high probability that if a child is important all its ancestors are also important [13].

The encoding of the geometry of the extracted connected tree structure saves bits as only optimal cut is to be encoded. There are two types of information 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 graylevels. 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 [8] have the most contribution in the approximation so each root wavelet is encoded. The encoding process is applied repeatedly for each of the geometric wavelet tree nodes in each tile.

#### 3.3.1 Encoding Geometry of the Support of the Wavelet:
The following information is encoded for each of the participating node \( \Omega \):

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

Left child and right child are defined as the sets of the pixels satisfying the inequality \( y - \tan \phi \cdot x \leq c \) and \( y - \tan \phi \cdot x \geq c \), 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 \( \Omega \) is participating in the sparse representation, then this event is encoded by using an additional bit. In case node is not a leaf node, then the indices of the parameter \( \phi \) and \( c \) of the bisecting line are encoded using the variable length coding [11].

3.3.2 Encoding the Coefficients of the Wavelet Polynomial:

The coefficients of the wavelet polynomial are quantized and encoded using the orthonormal basis. A bit allocation scheme [15] is applied depending upon the distribution functions of the coefficients of the wavelets participating in the sparse representation. 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 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.

After the encoding step, a rate distortion optimization process [14] is carried out in order to attain the desired bit rate. Pruning iterations [17] are applied, where at each iteration, the leaf node with minimal R-D slope is pruned until the desired rate is attained.

3.4 Decoding

In decoding, the compressed bit stream is read to find whether the participating node is the leaf node, has 1 child or 2 children [16]. 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 indexes of \( \phi \) and \( c \) are decoded and using these index parameters \( \phi \) and \( c \) of optimal cut are calculated. Thereafter, using this optimal cut, domain is partitioned into two subdomains [17]; 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.

4. Experimental Results

The proposed algorithm is tested on the still image of Lena of bit depth 8 and of size 512x512. 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” [18]. MSE and PSNR are given by the following relations:

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

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

where \( m \times n \) is the image size, \( x_{i,j} \) is the initial image and \( y_{i,j} \) is the reconstructed image. MSE and PSNR are inversely proportional to each other and higher value of the PSNR produces better image compression.
The performance of proposed method is compared against six algorithms. The PSNR values obtained by this method for the Lena image are compared with those obtained by the EZW [19], the SPIHT [20], the EBCOT [21], and the Bandelets [22] algorithms. Data presented in Table I show that the proposed method outperforms the EZW, the SPIHT, the EBCOT, and the Bandelets methods at low bit rates. But at high bit rates (0.125 bpp), the EBCOT algorithm performs better.

Table 1: PSNR Values in Decibels Compared with other State-of-the-Art Algorithms for test image, Lena

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>256:1</td>
<td>0.0315</td>
<td>25.38</td>
<td>26.1</td>
<td>-</td>
<td>-</td>
<td>26.87</td>
</tr>
<tr>
<td>128:1</td>
<td>0.0625</td>
<td>27.54</td>
<td>28.38</td>
<td>28.30</td>
<td>-</td>
<td>28.56</td>
</tr>
<tr>
<td>64:1</td>
<td>0.125</td>
<td>30.23</td>
<td>31.10</td>
<td>31.22</td>
<td>30.63</td>
<td>31.17</td>
</tr>
</tbody>
</table>

The proposed method reports a gain of 0.18 dB over the SPIHT method, 0.26 dB over the EBCOT method and 1.02 dB over EZW algorithm at the compression ratio of 128:1 for the Lena image. Figure 4 shows the reconstructed image of Lena, at a bit rate of 0.0625 bpp and PSNR 28.56.

![Original Lena](image1.jpg) ![Reconstructed Lena](image2.jpg)

Figure 4: (a) Original Lena, (b) Reconstructed Lena using the proposed method, 0.0625 bpp, PSNR=28.56.

5. Conclusion

In this correspondence, an efficient hybrid image compression algorithm using binary space partitioning scheme and geometric wavelets, well suited at low bit rates is proposed. The presented method produces the PSNR values that are competitive with the state-of-art coders in literature. 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). The algorithm is found to be extremely complex in computation and has high execution time. This makes the technique of image coding less practically applicable. The time complexity of the algorithm is analyzed in [23]. In future, new methods to reduce the execution time may be explored. The design of new “geometric” context modelling schemes combined with arithmetic encoding, may further improve the performance of the algorithm.
References


