Journal of
Electronic Imaging

JElectronicImaging.org

Modified set partitioning in hierarchical trees algorithm based on hierarchical subbands

Humberto de J. Ochoa Domínguez
Osslan O. Vergara Villegas
Vianey G. Cruz Sanchez

SPIE.
IS&T
imaging.org

Journal of Electronic Imaging 24(3), 033004 (May/Jun 2015)

Modified set partitioning in hierarchical trees algorithm based on hierarchical subbands

Humberto de J. Ochoa Domínguez,* **Osslan O. Vergara Villegas, and Vianey G. Cruz Sanchez**
Universidad Autónoma de Ciudad Juárez, Signal Processing Group, Electrical and Computer Engineering Department,
Av. del Charro 450 Norte, Ciudad Juárez, Chihuahua, 32310 México

Abstract. This paper introduces a modified set partitioning in hierarchical trees (SPIHT) algorithm that reduces the number of comparison operations and, consequently, the execution time needed to encode an image as compared to the SPIHT algorithm. The threshold of each independent subband is calculated after applying the discrete wavelet transform to the image. Scanning of the sets inside the subbands is determined by the magnitude of the thresholds that establishes a hierarchical scanning not only for the set of coefficients with larger magnitude, but also for the subbands. The algorithm uses the set partitioning technique to sort the transform coefficients. Results show that the modified SPIHT significantly reduces the number of operations and the execution time without sacrificing visual quality and the PSNR of the recovered image. [DOI: 10.1117/1.JEI.24.3.033004]

Keywords: image coding; transform coding; image processing; embedded coding; set partitioning in hierarchical trees.

Paper 14692 received Nov. 7, 2014; accepted for publication Apr. 14, 2015; published online May 8, 2015.

1 Introduction

It has been proven that image compression algorithms based on the discrete wavelet transform (DWT) provide high coding efficiency for natural images.[1–6] DWT has desirable properties, such as efficient multiresolution representation, scalability, vanishing moments, decorrelation, energy compaction, and embedded coding with progressive transmission, which are suitable for image compression.[7]

Since a few years ago, there has been a growing interest in the set partitioning in hierarchical trees (SPIHT) coder that is an improved version of the embedded zerotree wavelet algorithm.[8–12] SPIHT-based image compression can achieve rate scalability and high coding performance.[13–20] The encoder makes use of progressive transmission to exploit the self-similarity of the DWT coefficients across different scales. In addition, to iteratively extract the most important information in each subband, SPIHT uses partial ordering by comparing the transformed coefficient's magnitudes with a set of octave decreasing thresholds.

The coding performance of SPIHT follows close to that of the embedded block coding with optimized truncation, which forms the basis of the JPEG 2000 standard. However, SPIHT has much lower computational complexity. Therefore, there is a great interest to improve its performance.[17,21,22] Besides, it provides progressive image transmission, a fully embedded coded file, a simple quantization algorithm, lossless compression, and exact bit rate coding. Furthermore, SPIHT is very useful for applications where the user can quickly inspect the image and decide if it should be downloaded, is good enough to be saved, or needs refinement.

Even when SPIHT is not a standard, it is used in many video applications that use image compression with the embedded coding property[23–26] and medical imaging applications with SPIHT as a core.[8,20,27,28]

One of the main drawbacks of SPIHT is the slow coding speed owing to the dynamic processing order that depends on the image contents.[23,29–31] Efforts to improve the performance have been mostly concentrated in modifying the lists used to store the sets coordinates. For example, Pan et al.[21] proposed a listless modified SPIHT by taking advantage of the lifting wavelet scheme and weighted the transform coefficients according to the human visual system. Perceptual significance reordering allows higher peak signal-to-noise ratio (PSNR) values and better visual-quality images. The system reduces the complexity by suppressing the need of lists, but introduces a context-based arithmetic coder to the bit stream to gain more compression. However, substantial improvement is shown only for very low bit rates.

The work presented by Fang et al.[22] introduces a new interpolation-based lifting wavelet scheme combined with the Lagrange interpolation to transform the image. The technique also uses lists but considers coefficients higher than the threshold of four as important. Therefore, if D-sets in the list of insignificant sets (LIS) are important, the scanning procedure of the SPIHT is modified to reduce the unnecessary scans. Otherwise, traditional SPIHT is used. However, for high bit rates, the algorithm does not assure to have the same performance as that of SPIHT.

Set partitioning in hierarchical frequency bands (SPHFB)[17] uses the set partitioning technique to sort the transform coefficients. The threshold of each subband is calculated, and the subbands scanning sequence is determined by the magnitude of the thresholds. Therefore, the scanning sequence depends on the image content. Previous knowledge of the maximum bit depth of the subbands allows the encoder to find fast the subbands with more energy.

In SPIHT, the main task during a sorting pass is to find significant coefficients. Each coefficient in a subband has to

*Address all correspondence to: Humberto de J. Ochoa Domínguez, E-mail: hochoa@uacj.mx be compared with a threshold even though all the coefficients inside the subband are not significant to the threshold. This search could exacerbate the coding time. Therefore, reducing the number of comparison operations, during sorting passes, is important to reduce the coding time.

In this paper, a modified SPIHT algorithm for image coding, which combines the SPIHT lists and hierarchical subband scanning to save comparison operations during sorting passes, is proposed. The threshold of each independent subband is calculated to find its maximum depth in bits. The sets inside a subband are scanned according to the magnitude of its subband threshold. The scanning of a spatial orientation tree stops if the threshold of a subband containing descendants is less than the current threshold. Hence, sets inside subbands with a higher threshold are scanned first. Two major modifications are proposed: (1) the calculation of one threshold per subband to impose a hierarchical scanning of sets with higher energy and, in contrast to the SPIHT, (2) the LIS must be initially empty and is populated with coordinates of sets in the subbands of higher hierarchy to avoid unnecessary entries to the list. The two modifications allow to outperform the performance in time of SPIHT, listless modified (LM)SPIHT, and Fang et al.[22] methods. Along the text, the terms method and algorithm are used indistinctly.

The remainder of the paper is organized as follows. Section 2 gives a theoretical model of the comparison operations of SPIHT. Section 3 describes the modified SPIHT. Section 4 shows the experimental results. Conclusions are drawn in Sec. 5.

2 Modeling Comparison Operations of SPIHT

SPIHT operates on a hierarchical pyramidal structure χ that is produced by transforming an image by using a DWT implemented with a filter bank.[14] The energy compaction is one of the most important metrics in the evaluation of filters used in the transform coding schemes. It refers to the property of how the energy in the transform domain is distributed among its coefficients. For encoding purposes, it is required that most of the energy of an image is concentrated in a few coefficients. In this work, the 9-7 tap biorthogonal wavelet filter bank—Cohen-Daubechies-Feauveau (CDF 9/7)—is used because it is better than others in terms of energy compaction.[7] After transformation, the highest-frequency subband has the finest samples and lies at the bottom right of the pyramid. The lowest-frequency subband has the coarsest samples and lies at the top left of the pyramid. At each scale l, the three high-frequency subbands capture the horizontal, vertical, and diagonal directions. The last scale of decomposition yields four subbands including the lowest-frequency subband. Therefore, if the image is decomposed using L scales, a total of $3L+1$ subbands are obtained.

At the beginning of the SPIHT encoding process, the maximum threshold $\text{Th}_{\max}$ of χ is calculated based on the number of bits needed to represent the highest magnitude coefficient $\text{cs}(i,j)$, at position (i,j), by using Eq. (1):

$$\text{Th}_{\max} = \left\lfloor \log_2(\max_{(i,j)\in\chi}\{|\text{cs}(i,j)|\}) \right\rfloor, \tag{1}$$

where $|.|$ and $\lfloor . \rfloor$ are the absolute value and the floor operations, respectively. $\text{Th}_{\max}$ becomes the first current threshold Th_c and is the only initial information, about

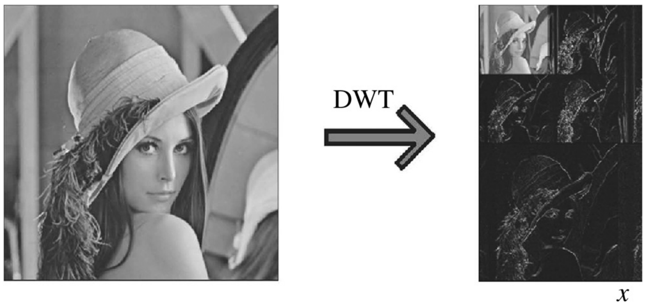


Fig. 1 Pyramidal structure of the Lena image after wavelet decomposition with $L=2$.

the magnitude of all the coefficients, that is sent to the decoder. Figure 1 shows an example of the pyramidal structure for $L=2$ of the Lena image with seven subbands and a resulting $\text{Th}_{\max}=9$.

The data structure proposed by Said and Pearlman[14] allows the wavelet coefficients to be arranged into 2×2 arrays that are offsprings of a coefficient of a lower-resolution subband, except for the coefficients in the coarsest subband, where all except one of them are root nodes of a spatial orientation tree. To give a more realistic view, the threshold of each independent subband of Fig. 1 can be calculated to plot the three-dimensional structure of Fig. 2. The arrows show how the various levels of the trees are related because of the similarity between the subbands. The numbers represent the thresholds in bits. Observe the different distances between the subbands.

The trees are further partitioned into four types of sets of coordinates of coefficients as detailed next.

- $\mathcal{O}(i,j)$: set of coordinates of the four offsprings of node (i,j). For example, in Fig. 2, the set $\mathcal{O}(1{,}0)$ consists of coordinates of the coefficients d_1, d_2, d_3, and d_4.
- $\mathcal{D}(i,j)$: set of coordinates of the descendants of node (i,j). For example, in Fig. 2, the set $\mathcal{D}(1{,}0)$ consists of coordinates of the coefficients $d_1,\ldots,d_4,\ldots,d_{11},\ldots,d_{44}$.

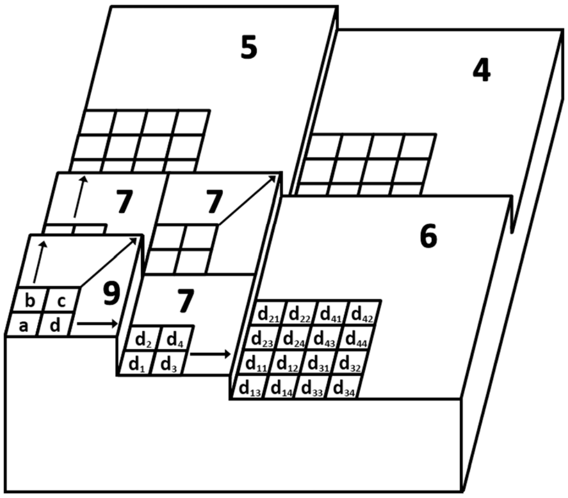


Fig. 2 Three-dimensional plot of the pyramidal structure χ of the Lena image after wavelet decomposition with $L=2$ (seven subbands) showing the spatial orientation trees. The numbers represent the threshold's magnitude of each subband in bits.

- $\mathcal{H}$: set of coordinates of the roots of all spatial orientation trees (essentially 3/4 of the coefficients in the coarsest subband).
- $\mathcal{L}(i,j) = \mathcal{D}(i,j) - \mathcal{O}(i,j)$. For example, in Fig. 2, the set $\mathcal{L}(1{,}0)$ consists of coordinates of the coefficients $d_{11}, \ldots, d_{44}$.

Two important passes, the sorting and refinement passes, are defined. In a sorting pass, the coefficients are compared with Th_c. A "1" bit followed by a sign bit is sent to the decoder for every coefficient equal to or larger than Th_c. In addition, the coefficient is marked as significant and it will not be scanned again. However, it will be refined in subsequent refining steps. If the coefficient is less than the current threshold, a "0" is sent to the decoder. It is said that the coefficient is insignificant and it will be compared with the next current thresholds in further sorting passes until it is found significant or the encoding process ends. The operation to determine the significance of a coefficient during a sorting pass is called a comparison operation. The sorting passes also divide the set of samples into partitioning subsets of samples. Each coefficient in a set is compared against the current threshold. If no coefficient is significant, a 0 is sent to the decoder to indicate that the entire set is insignificant to that specific threshold.

From Fig. 2, it can be seen that sets of type $\mathcal{O}(i,j)$, inside the three directional subbands, at scale 2 may be found significant until the third sorting pass, when $\mathrm{Th}_c = 7$. Also, the first significant sets of type $\mathcal{L}(i,j)$ in the horizontal, vertical, and diagonal subbands may be found significant until the fifth, fourth, and sixth passes, respectively. Notice that the encoding process is blind beyond the current threshold. That is, no information of the subbands is available to the process. Hence, all the descendants in a tree must be tested for significance, even though they belong to subbands with a threshold less than that of the current threshold, thus performing unnecessary comparison operations during a sorting pass. According to Fig. 2, for an image size of $M \times N$, the maximum number of unnecessary comparison operations C in the first sorting pass can be approximated by Eq. (2):

$$C = M \times N \sum_{l=0}^{L-1} \frac{1}{2^{2(L-l)}} \sum_{k=1}^{3} I_{l,k}, \tag{2}$$

$$I_{l,k} = \begin{cases} 1, & \mathrm{Th}_c > \mathrm{Th}_{l,k} \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

where $\mathrm{Th}_{l,k}$ is the threshold of the subband at scale l and direction k. $I_{l,k}$ indicates subbands with a threshold less than Th_c. However, as the number of passes increase, according to the bit rate requirements, Eq. (3) becomes

$$I_{l,k} = \begin{cases} \mathrm{Th}_{\max} - \mathrm{Th}_{l,k}, & \mathrm{Th}_c > \mathrm{Th}_{l,k} \\ 0, & \text{otherwise} \end{cases}. \tag{4}$$

Equation (4) is an ill-conditioned equation because the number of unnecessary comparison operations depend on the distance of each subband threshold to the maximum threshold. For natural images, subbands with higher resolution have a lower threshold.[14] Therefore, the higher the bit rate, the more are the unnecessary operations. Further, there is an increase of the number of bits required to signal the set's significance, which only increases the bit rate without contributing to an increase in image quality.

3 Modified SPIHT

In Sec. 2, it was shown that the use of only one threshold leads to a blind encoder. This increases the number of comparison operations during a sorting pass. One way to reduce the number of operations is by inverting the inequality of Eq. (3). Hence, the contribution during the scanning process is only from the subbands with a threshold higher than or equal to the current threshold as shown in Eq. (5):

$$I_{l,k} = \begin{cases} 1, & \mathrm{Th}_c \le \mathrm{Th}_{l,k} \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

Algorithm 1 defines four steps: (1) initialization, (2) sorting pass, (3) refinement pass, and (4) threshold quantization step.

Here, besides the maximum threshold, it is necessary to obtain information about the maximum height of each subband. Therefore, in the modified SPIHT, the encoder calculates the subbands threshold by using Eq. (6) and sends them to the decoder in an orderly manner according to the subband that they represent.

$$\mathrm{Th}_{l,k} = \left\lfloor \log_2\!\left(\max_{(i,j)\in s_{l,k}}\{|\mathrm{cs}_{l,k}(i,j)|\}\right) \right\rfloor, \tag{6}$$

where $s_{l,k}$ is the k'th subband at scale l, and $\mathrm{cs}_{l,k}$ is a coefficient inside the subband. Hence, a short register containing the $3L+1$ thresholds must be initialized, with the current threshold Th_c initialized to the maximum threshold. It is important to note that, in the initialization stage, the modified algorithm inspects orderly each subband of χ in order to determine the thresholds, contrary to the SPIHT, which inspects all coefficients of χ to compute a single threshold. Hence, the time added by Eq. (1) to the SPIHT process is the same as the time added by Eq. (6) to the modified algorithm process.

Three lists are maintained as in SPIHT: the list of significant pixels (LSP), the list of insignificant pixels (LIP), and the list of insignificant sets (LIS). LIP contains the coordinates of coefficients that have not been significant to the current or previous sorting passes and is initialized with the set $\mathcal{H}$. The encoding process begins by examining the coordinates in LIP. Coefficients are tested for significance by using Eq. (7):

$$2^{\mathrm{Th}_c+1} > |\mathrm{cs}_{l,k}(i,j)| \ge 2^{\mathrm{Th}_c}. \tag{7}$$

If the coefficient is insignificant, a 0 is sent to the bit stream leaving the LIP unchanged. If the coefficient is significant, a 1 is sent to the bit stream followed by a sign bit obtained by Eq. (8). Then, the coefficient position is removed from the LIP and placed in the LSP. LSP contains the coordinates of coefficients found significant during sorting passes and is initially empty.

Algorithm 1 Modified SPIHT

```
1: Initialization: output $\mathrm{Th}_{l,k} = \log_2 \left\lfloor (\max_{(i,j)\in s_{l,k}}\{|\mathrm{cs}_{l,k}(i,j)|\}) \right\rfloor$. Set $\mathrm{Th}_c = \max\{\mathrm{Th}_{l,k}\}$. Set LSP $= \{\varnothing\}$, LIS $= \{\varnothing\}$. Add the coordinates $(i,j) \in \mathscr{H}$ to LIP.
2: Sorting pass:
  1. For each entry $(i,j)$ in the LIP do:
    (a) Output $\mathscr{S}_{l,k}(i,j)$;
    (b) If $\mathscr{S}(i,j) = 1$, then move $(i,j)$ to LSP and output $\delta[cs_{l,k}(i,j)]$
  2. For each subband threshold $\mathrm{Th}_{l,k}$ do:
    (a) If $\mathrm{Th}_{l,k} = \mathrm{Th}_c$, then allow sets of subband $s_{l,k}$ to be added to the LIS.
  3. For each entry $(i,j)$ in the LIS do:
    (a) If the set is of type A ($2 \times 2$ coefficients), then
      i. Output $\mathscr{S}_{l,k}[\mathscr{D}(i,j)]$;
      ii. If $\mathscr{S}_{l,k}[\mathscr{D}(i,j)] = 1$, then
        • For each $\mathrm{cs}_{l,k}(i,j) \in \mathscr{D}(i,j)$ do
          - Output $\mathscr{S}_{l,k}(i,j)$
          - If $\mathscr{S}_{l,k}(i,j) = 1$, then move the position of $cs_{l,k}(i,j)$ to LSP and output $\delta[cs_{l,k}(i,j)]$
          - If $\mathscr{S}_{l,k}(i,j) = 0$, then move the position of $cs_{l,k}(i,j)$ to LIP
        • Remove $\mathscr{D}(i,j)$ from LIS
    (b) If the set is of type B ($4 \times 4$ or more coefficients), then
      i. Output $\mathscr{S}_{l,k}[\mathscr{L}(i,j)]$;
      ii. If $\mathscr{S}_{l,k}[\mathscr{L}(i,j)] = 1$, then divide $\mathscr{L}(i,j)$ into four subsets and add the result at the end of LIS;
      iii. Remove $\mathscr{L}(i,j)$ from LIS.
3: Refinement pass: For each entry $(i,j)$ in the LSP, except those included in the last sorting pass, output the $\mathrm{Th}_c$'th bit of $|\mathrm{cs}_{l,k}(i,j)|$
4: Quantization step: Decrement $\mathrm{Th}_c$ by 1 and go to step 2.
```

$$\delta(i,j) = \begin{cases} 0, & \mathrm{cs}_{l,k}(i,j) \geq 0 \\ 1, & \mathrm{cs}_{l,k}(i,j) < 0 \end{cases}. \tag{8}$$

LIS stores the coordinates of the roots of sets of type $\mathcal{D}$ or $\mathcal{L}$ that have not been significant to either the current or previous passes. Significance of a set $\mathcal{T}$ of type $\mathcal{D}$ or $\mathcal{L}$ is tested by using Eq. (9):

$$\mathcal{S}_{l,k}\{\mathcal{T}\} = \begin{cases} 1, & \max_{(i,j)\in\mathcal{T}}\{|\mathrm{cs}_{l,k}(i,j)|\} \geq 2^{\mathrm{Th}_c} \\ 0, & \text{otherwise} \end{cases}. \tag{9}$$

After examining LIP, the list of thresholds is examined. If the subband threshold is less than Th_c, the entire subband is insignificant and it is skipped. Otherwise, the subband is significant to the current threshold, and it is allowed to place the sets at the end of LIS to be inspected in the current pass, as follows:

If a set at coordinate (i,j) is significant, a 1 is transmitted. What we do next depends on whether the set is of type $\mathcal{D}$ or $\mathcal{L}$ as in SPIHT.

If set is of type $\mathcal{D}$, the four descendants $\mathcal{O}(i,j)$ are tested; a 1 is transmitted for each significant coefficient, followed by its sign bit. Their coordinates are placed in the LSP. A 0 is transmitted for each insignificant coefficient and the coordinate is added to the LIP. $\mathcal{O}(i,j)$ is removed from the LIS, and if the set $\mathcal{L}(i,j)$ exists, it must be moved to the end of the LIS.

If set is of type $\mathcal{L}$, each coordinate in $\mathcal{O}(i,j)$ is added to the end of the LIS to be treated as the root of a set of type $\mathcal{D}$.

In the modified SPIHT, the LIS is initially empty and is populated with coordinates of sets according to the thresholds of the subband to which they belong. Notice that SPIHT initializes the LIS with the elements of $\mathscr{H}$ that have descendants, including those inside subbands with a threshold less than the maximum threshold.

Once the LIP and LIS have been processed, the sorting pass ends and starts the refinement step. In the refinement pass, the coefficients in the LSP, resulting from the previous passes, are examined and the output is the $\mathrm{Th}_c^{\mathrm{th}}$ most significant bit of the coefficients in the coordinates of the list. Before starting a new sorting pass, Th_c is decremented. The bit rate is controlled by counting the number of bits of the bit stream and by comparing the count with the total bit budget $M \times N \times \text{Bit rate}_{\text{bpp}}$.

The pseudocode of the modified algorithm is shown below. Notice that after inspecting the LIP, the thresholds register is inspected to decide whether to add the subband sets to the LIS, which means that if there are no subband thresholds equal to the current threshold, the only sets inspected are those in the subbands with a threshold greater than Th_c that were found in previous passes.

4 Experimental Results

In this section, the performance of the proposed modified SPIHT is examined and compared with SPIHT, SPHFB, LMSPIHT, and Fang et al.[22] methods using the images of Lena, Barbara, and Mandrill at 8 bpp monochrome 512×512 size.[32] The pyramidal decomposition was obtained by the CDF 9/7 filter bank[33] with a reflection extension to each image and $L = 6$. The bit rate interval tested was from 0.01 to 4 bpp. No entropy coder was used on the resulting bit stream. PSNR, visual quality, performance with respect to the number of comparison operations in sorting passes, and time performance were compared with those obtained with SPIHT (Ref. 34) and with other algorithms. The methods used in this paper were implemented in C language under the Ubuntu operating system, running on a Dell precision T7500 personal computer, with an Intel Xeon quadcore clocked at 2.40 GHz and 8 GB of memory. It is important to mention that the only processes running in the CPU was the method to be analyzed (one at a time). The comparison operations were counted during the sorting passes, and the execution time was taken including the initialization stage up to the end of the encoding process.

4.1 *Comparison Operations*

Table 1 shows the total number of comparison operations after each sorting pass for Lena, Barbara, and Mandrill images. Observe that the modified SPIHT starts from a reduced number of comparison operations. Also, note that the number of comparisons to determine whether a subband is significant is negligible. For example, with $L = 6$, the number of comparisons of Th_c with each threshold in the register of thresholds is 19 $(3L + 1)$. Even if this number is added to the results of the modified SPIHT in Table 1, the ratio of comparison operations between SPIHT and the modified SPIHT remains.

Table 1 Total number of comparison operations versus sorting passes for Lena, Barbara, and Mandrill.

	Lena		Barbara		Mandrill	
Sorting pass	Modified SPIHT	SPIHT	Modified SPIHT	SPIHT	Modified SPIHT	SPIHT
1	64	262,144	64	262,144	64	262,144
2	166	786,406	168	786,408	158	786,398
3	381	1,572,797	370	1,572,786	252	1,572,732
4	1775	2,621,487	2057	2,621,641	522	2,621,194
5	10,185	3,932,825	9599	3,933,199	6147	3,932,515
6	66,134	5,507,718	71,188	5,509,492	69,434	5,509,162
7	233,237	7,349,813	463,724	7,370,524	521,501	7,386,749
8	656,261	9,461,525	1,303,794	9,544,818	1,678,328	9,577,864
9	1,679,103	11,848,703	2,687,360	12,020,512	3,469,754	12,094,394
10	3,246,630	14,525,158	4,612,885	14,790,133	5,727,595	14,929,579
11	5,394,682	17,541,498	6,999,252	17,850,484	8,289,274	18,015,098
12	8,044,440	20,934,968	9,790,561	21,216,897	11,000,790	21,248,566
13	10,906,275	24,535,459	12,772,826	24,767,066	13,723,921	24,491,770
14	13,867,081	28,234,921	15,804,936	28,360,124		

Note: SPIHT, set partitioning in hierarchical trees.

To complete the first sorting pass, for the Lena, Barbara, and Mandrill images, SPIHT takes 262,144 comparison operations, while the modified SPIHT takes only 64. As the number of sorting passes increases, the number of comparisons is exacerbated when using SPIHT, because the number of operations accumulates in every pass. For example, to complete eight sorting passes, SPIHT takes ∼9.5 millions of comparison operations for the Lena, Barbara, and Mandrill images, while the modified SPIHT takes only 656,261 for the Lena, 1,303,794 for the Barbara, and 1,678,328 for the Mandrill. Finally, after the 14th sorting pass, SPIHT takes 2̃8.4 millions of comparison operations for the Lena and Barbara, while the modified SPIHT takes 13.8 and 15.8 millions, respectively. In the case of Mandrill, SPIHT takes 2̃4.5 millions of comparison operations, while the modified SPIHT takes 13.7 millions in 13 sorting passes.

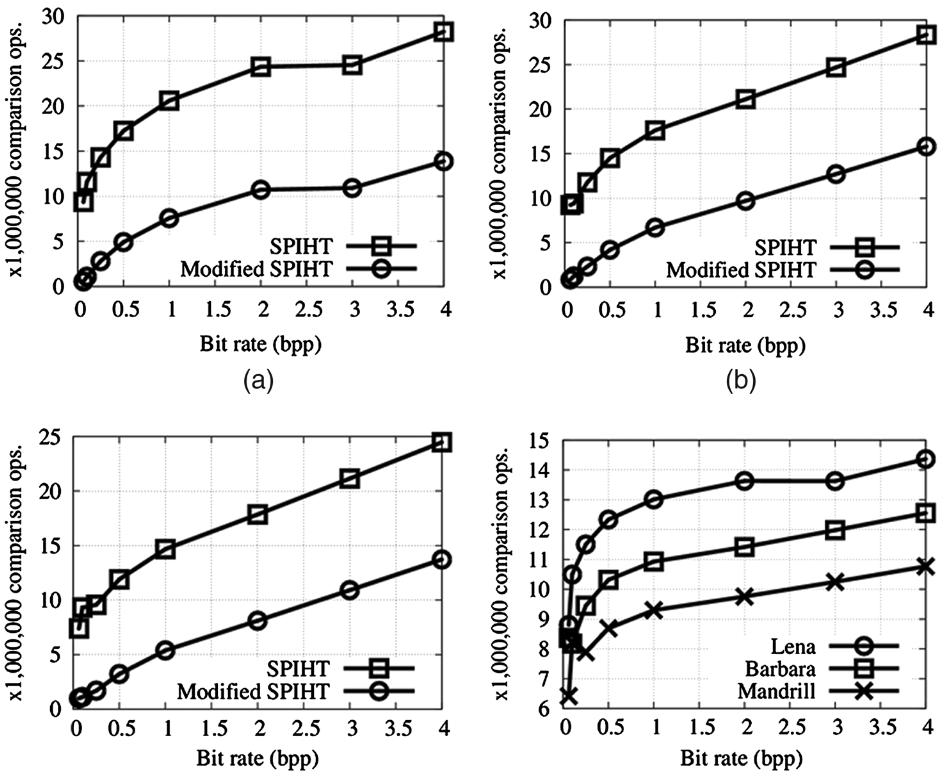
Figures 3(a)–3(c) show the total number of comparison operations for Lena, Barbara, and Mandrill versus the bit rate. For the sake of visual clarity, the bit rates of 0.01 and 0.75 bpp are not plotted. It is interesting to note that for the Lena, SPIHT takes almost the same number of comparison operations to reach 0.0625 bpp as the modified SPIHT to reach ∼1.7 and 2 bpp in the cases of Barbara and Mandrill, respectively.

Table 2 shows the number of sorting passes needed to obtain a bit rate. Observe that it is necessary to spend at least six sorting passes to reach 0.01 bpp. Because of the embedded characteristic of the code, the current pass depends on previous passes. Therefore, SPIHT starts from a high number of comparison operations since the first sorting pass, while the modified SPIHT starts from a minimum of these operations.

Table 2 Bit rate versus number of sorting passes.

	Lena		Barbara		Mandrill	
Bit rate (bpp)	Modified SPIHT	SPIHT	Modified SPIHT	SPIHT	Modified SPIHT	SPIHT
0.01	6	6	6	6	6	6
0.0625	8	8	8	8	8	7
0.1	9	9	8	8	8	8
0.25	10	10	9	9	8	8
0.5	11	11	10	10	9	9
0.75	11	11	10	10	10	10
1.0	12	12	11	11	10	10
2.0	13	13	12	12	11	11
3.0	13	13	13	13	12	12
4.0	14	14	14	14	13	13

Figure 3(d) shows the comparison operations saved for each image. Lena exhibits the highest saving even though from 2 to 3 bpp the plot has an almost constant behavior



(a) (b) (c) (d)

Fig. 3 Number of comparison operations versus bit rates for (a) Lena, (b) Barbara, and (c) Mandrill images. (d) Plot of comparison operations saved.

owing to the fact that most of the encoded bits belong to refinement passes. In the cases of Barbara and Mandrill, there is an almost linear behavior from 1 to 4 bpp. Notice that the comparison operations gain exhibited in the plots of Figs. 3(a), 3(b), and 3(c) approximates to 2:1 as the bit rate approaches to 4 bpp.

Previous works, such as LMSPIHT[21] and that reported by Fang et al.,[22] do not consider the bit depth of each subband and the search for significant coefficients is carried out in subbands with thresholds less than the current threshold [see Eqs. (3) and (4)], performing about the same number of comparison operations during a sorting pass as in SPIHT. In LMSPIHT, the partitioning structure is modified causing the length of the LIP and LIS lists to be reduced during the first sorting pass. Fang et al.[22] consider the case when the current threshold is too large or too small, the sets $\mathcal{O}(i,j)$ will be not so important. This improves the speed of compression and the ratio of important coefficients that convey information about the coded image in the bit stream. However, only for low bit rates, the PSNR is the same as SPIHT. The work that considers the hierarchical scanning of subbands is the SPHFB,[17] which makes use of Eq. (5), resulting in about the same number of comparison operations as the modified SPIHT.

4.2 *Execution Time*

The aim of the modified SPIHT is to reduce the number of comparison operations needed to encode an image and, hence, the number of searches of significant coefficients. This reduction has a direct impact on the execution time that is at least halved as compared to the SPIHT.

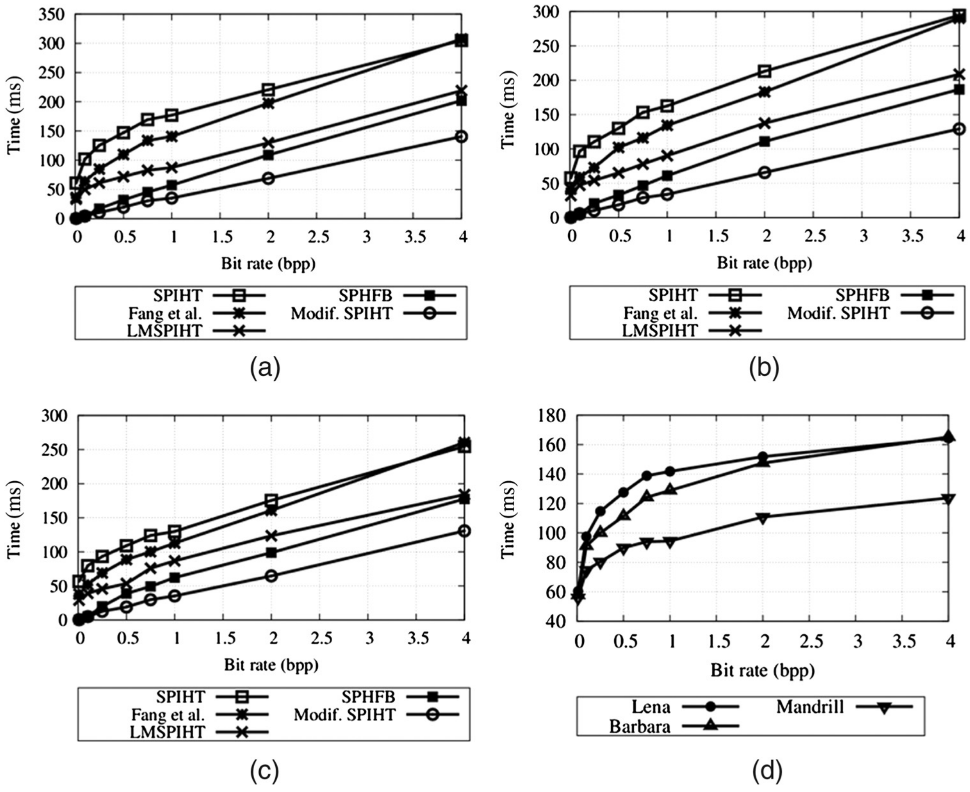
Figure 4 shows the plots of the execution time of the modified SPIHT, compared with SPIHT, SPHFB, LMSPIHT, and Fang et al. methods for the Lena, Barbara, and Mandrill images. Additionally, the plot of time saving of the modified SPIHT with respect to SPIHT is depicted.

The numerical results for different bit rates are summarized in Table 3. The ratios SPIHT/modified SPIHT at bit rates of 0.01, 0.25, 0.5, 1, and 4 bpp for the Lena image are 147.65, 23.28, 11.73, 7.52, 5.02, and 2.16. The modified SPIHT takes only 0.67, 8.51, 13.28, 19.91, and 46.1% of the total time needed for SPIHT to encode the image, whereas for the same bit rates, SPHFB takes 0.67, 13.81, 21.91, 32.39, and 66.23%; LMSPIHT takes 57.74, 48.78, 48.92, 49.47, and 71.80%; and Fang et al.[22] takes 59.19, 65.59, 74.02, 79.42, and 100.8% of the total time taken by SPIHT.

For the Barbara image, the ratios are 205.26, 10.49, 6.9, 4.8, and 2.27, taking 0.48, 9.52, 14.35, 20.81, and 43.88% of the total time as compared to SPIHT, while SPHFB takes 0.48, 18.79, 25.33, 37.49, and 63.38%; LMSPHIT takes 55.49, 49.00, 50.22, 55.41, and 70.86%; and Fang et al.[22] takes 71.92, 65.72, 78.45, 85.52, and 98.66% compared with the time taken by SPIHT.

For the Mandrill image, the resulting ratios are 98.7, 7.26, 5.68, 3.67, and 1.94, taking only 1.01, 13.75, 17.58, 27.22, and 51.38% of the total time needed by SPIHT, whereas SPHFB takes 1.01, 21.49, 35.67, 47.91, and 69.78%; LMSPIHT takes 51.52, 48.98, 48.98, 66.80, and 72.27%; and Fang et al.[22] takes 67.21, 74.02, 81.24, 86.25, and 102.25% of the time taken by SPIHT.

Notice that in SPIHT, the search for significant coefficients exacerbates the execution time. Hence, it is the most time-consuming operation in the coding process.



(a) (b) (c) (d)

Fig. 4 Execution time versus bit rates for (a) Lena, (b) Barbara, and (c) Mandrill images. (d) Time saved of the modified set partitioning in hierarchical trees (SPIHT) with respect to SPIHT.

Table 3 Execution time comparison (in milliseconds).

Bit rate (bpp)	Image	Modified SPIHT	SPIHT	SPHFB	LMSPIHT	Fang et al.[22]
0.01	Lena	**0.412**	60.83	0.412	33.30	36.01
	Barbara	**0.282**	57.88	0.282	32.12	41.63
	Mandrill	**0.573**	56.55	0.573	29.14	38.01
0.1	Lena	**4.378**	101.90	4.45	49.92	63.94
	Barbara	**5.54**	96.65	5.72	47.32	58.81
	Mandrill	**5.05**	79.74	5.53	39.06	51.89
0.25	Lena	**10.68**	125.37	17.32	61.16	84.75
	Barbara	**10.52**	110.52	20.77	54.16	72.64
	Mandrill	**12.82**	93.24	20.04	45.67	69.02
0.5	Lena	**19.53**	147.06	32.23	71.95	109.80
	Barbara	**18.66**	130.01	32.94	65.30	102.00
	Mandrill	**19.18**	109.11	38.92	53.45	88.645
0.75	Lena	**30.63**	169.40	45.4	82.62	133.53
	Barbara	**28.97**	153.15	46.69	78.08	116.12
	Mandrill	**29.90**	123.95	49.21	76.07	100.15
1.0	Lena	**35.25**	177.02	57.35	87.58	140.60
	Barbara	**33.87**	162.71	61.01	90.17	134.28
	Mandrill	**35.34**	129.83	62.21	86.73	112.76
2.0	Lena	**68.89**	220.67	108.93	129.73	197.40
	Barbara	**65.65**	213.06	110.94	137.47	183.09
	Mandrill	**64.61**	175.43	98.86	123.58	160.74
4.0	Lena	**140.53**	304.84	201.92	218.88	307.30
	Barbara	**129.18**	294.35	186.57	208.60	290.42
	Mandrill	**130.78**	254.53	177.63	183.97	260.28

Note: SPHFB, set partitioning in hierarchical frequency bands.
Bold values indicate the best result.

Table 4 Improvement in the speed of compression (in %) with respect to SPIHT.

Bit rate (bpp)	Image	Modified SPIHT	SPHFB	LMSPIHT	Fang et al.[22]
0.01	Lena	**99.32**	99.32	45.26	40.80
	Barbara	**99.51**	99.51	44.51	28.08
	Mandrill	**98.99**	98.99	48.47	32.79
0.1	Lena	**95.70**	95.63	51.01	37.25
	Barbara	**94.27**	94.08	51.04	39.15
	Mandrill	**93.67**	93.06	51.02	34.93
0.25	Lena	**91.48**	86.18	51.22	32.40
	Barbara	**90.48**	81.21	51.00	34.27
	Mandrill	**86.25**	78.51	51.02	25.98
0.5	Lena	**86.72**	78.08	51.07	25.34
	Barbara	**85.65**	74.66	49.77	21.54
	Mandrill	**82.42**	64.33	51.01	18.76
0.75	Lena	**81.92**	73.20	51.23	21.17
	Barbara	**81.08**	69.51	49.02	24.18
	Mandrill	**75.88**	60.30	38.63	19.20
1.0	Lena	**80.09**	67.60	50.53	20.57
	Barbara	**79.18**	62.50	44.58	17.47
	Mandrill	**72.78**	52.08	33.20	13.15
2.0	Lena	**68.78**	50.64	41.21	10.55
	Barbara	**69.19**	47.93	35.48	14.07
	Mandrill	**63.17**	43.65	29.56	8.37
4.0	Lena	**53.90**	33.76	28.20	−0.81
	Barbara	**56.11**	36.62	29.13	1.34
	Mandrill	**48.62**	30.21	27.72	−2.26

Note: Bold values indicate the best result.

Table 4 shows the improvement in speed of compression in % for each bit rate, computed on the time difference between SPIHT and each method for a given bit rate, and taking the time of SPIHT as 100%. In LMSPIHT, the percentage is quite constant upto 0.75 bpp, while Fang et al.[22] shows a greater decrease in performance than the rest of the methods as the bit rate increases. Notice that the modified SPIHT exhibits the highest improvement and SPHFB follows this method because both are based on hierarchical subbands.

The significance reordering of the coefficients in LMSPIHT allows to code more significant information earlier. However, for bit rates >0.5 bpp, this method lowers its performance because it scans subbands with thresholds above the current threshold. In SPHFB, the use of hierarchical subbands allows to increase the performance in time. However, the number of bits is increased because new bits are added to the bit stream to signal the set type reducing the PSNR as it will be seen in Sec. 4.3.

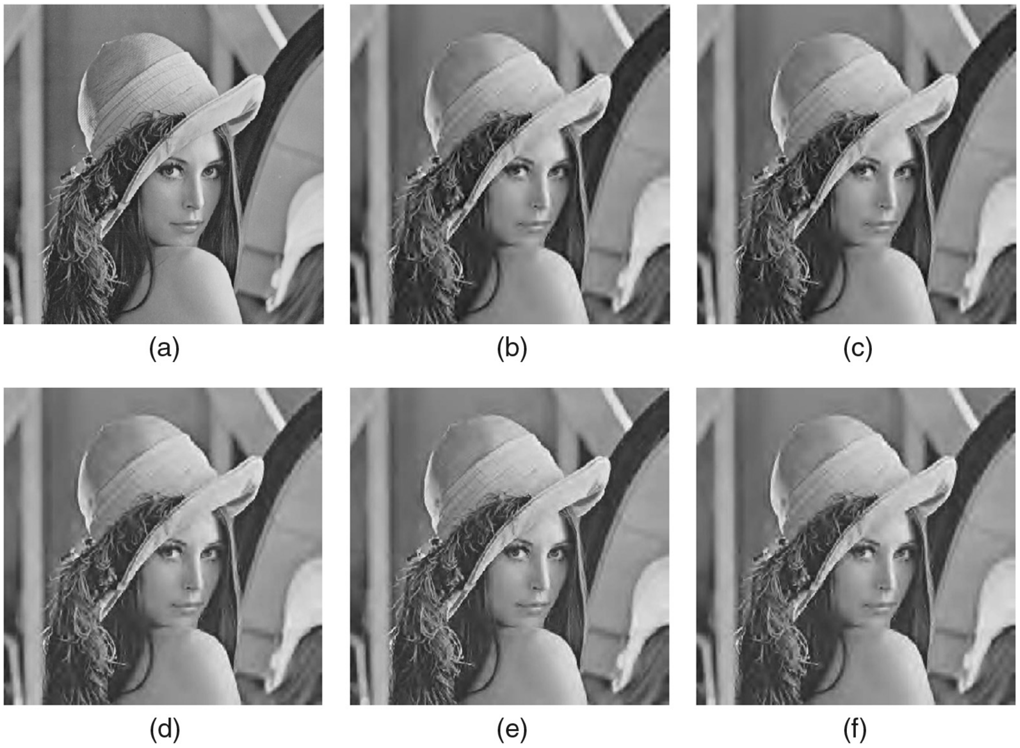


Fig. 5 Lena images obtained at a bit rate of 0.1 bpp: (a) original image, (b) SPIHT peak signal-to-noise ratio (PSNR) = 29.81 dB, (c) modified SPIHT PSNR = 29.82 dB, (d) set partitioning in hierarchical frequency bands (SPHFB) PSNR = 30.31 dB, (e) LMSPIHT PSNR = 31.21 dB, and (f) Fang et al. PSNR = 29.81 dB.

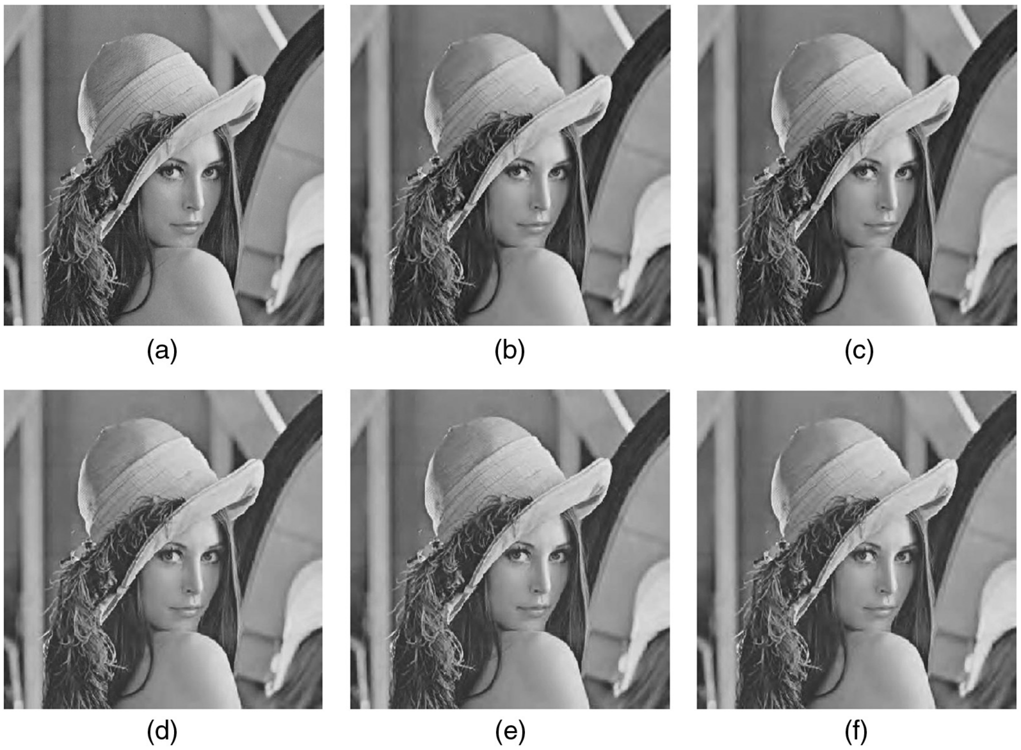


Fig. 6 Lena images obtained at a bit rate of 0.25 bpp: (a) original image, (b) SPIHT PSNR = 33.65 dB, (c) modified SPIHT PSNR = 33.65 dB, (d) SPHFB PSNR = 34.15 dB, (e) LMSPIHT PSNR = 34.88 dB, and (f) Fang et al. PSNR = 33.65 dB.

Fig. 7 Lena images obtained at a bit rate of 0.5 bpp: (a) original image, (b) SPIHT PSNR = 36.78 dB, (c) modified SPIHT PSNR = 36.78 dB, (d) SPHFB PSNR = 36.78 dB, (e) LMSPIHT PSNR = 36.80 dB, and (f) Fang et al. PSNR = 36.78 dB.

4.3 *PSNR Results*

The distortion between the original and recovered images was measured by using the PSNR formula:

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \mathrm{dB}. \tag{10}$$

MSE is the mean squared error between the original and the reconstructed image. Figure 5(a) shows the original image, Fig. 5(b) the reconstructed image by using SPIHT, Fig. 5(c) the modified SPIHT, Fig. 5(d) SPHFB, Fig. 5(e) LMSPIHT, and Fig. 5(f) Fang et al. at 0.1 bpp. The resulting PSNRs were 29.81 and 29.82, 30.31, 31.21, and 29.81 dB, respectively.

Figure 6 shows the original image [Fig. 6(a)] along with the reconstructed images at 0.25 bpp by using SPIHT [Fig. 6(b)] and the modified SPIHT [Fig. 6(c)], SPHFB [Fig. 6(d)], LMSPIHT [Fig. 6(e)], and Fang et al. [Fig. 6(f)]. The resulting PSNRs were 33.65, 33.65, 34.15, 34.88, and 33.65 dB, respectively.

Figure 7 shows the original image [Fig. 7(a)] along with the reconstructed images at 0.5 bpp by using the SPIHT [Fig. 7(b)] and the modified SPIHT [Fig. 7(c)], SPHFB [Fig. 7(d)], LMSPIHT [Fig. 7(e)], and Fang et al [Fig. 7(f)]. Notice that the PSNR yielded by most of the methods is 36.78 dB, except LMSPIHT whose PSNR is 36.80 dB.

Table 5 shows the comparison of PSNRs. Observe that the LMSPIHT yields, on average, higher PSNR for very low bit rates (≤0.25 bpp). For a bit rate of 0.5 bpp, only the Lena image has a PSNR of 0.02 dB above the other methods. On average, for bit rates >0.25 bpp, the modified SPIHT and SPIHT yield a higher PSNR. However, for bit rates >1 bpp, the modified SPIHT performs slightly better because of Eq. (5).

4.4 *Visual Quality*

The visual quality assessment of the recovered images was addressed according to the structural similarity index measure (SSIM).[35] The closer the index to 1, the more similar the two compared signals. The recovered images were compared with the original one.

Table 6 compares the SSIM. For very low bit rates (≤0.25 bpp), LMSPIHT has a slight gain over the rest of the methods. For bit rates >0.5 bpp, the modified SPIHT and the SPIHT have a higher SSIM. Both methods perform similar to each other except for the Barbara and Mandrill images at high bit rates (>0.5 bpp) where the modified SPIHT performs slightly better.

5 Conclusions

In this paper, a modified SPIHT algorithm that combines the advantages of SPIHT and the hierarchical scanning of the subbands is proposed. Energy compaction and ordering are two essential requirements for good compression. The DWT CDF 9/7 fulfilled the first requirement and the second requirement was improved by scanning the subband sets according to their corresponding hierarchy imposed by the thresholds, thereby reducing the entries to the LIS, making it grow adaptively according to the image characteristics and drastically reducing the number of comparisons and the time taken by the encoder per sorting pass. The modified SPIHT outperforms the performance in time of SPIHT and other methods analyzed. Another advantage is that the use of the subband thresholds ensures that higher magnitude

Table 5 Comparison of peak signal-to-noise ratio (in dB).

Bit rate (bpp)	Image	Modified SPIHT	SPIHT	SPHFB	LMSPIHT	Fang et al.[22]
0.01	Lena	22.46	22.54	24.80	**25.70**	22.54
	Barbara	20.09	20.02	22.53	**23.30**	20.02
	Mandrill	19.22	19.19	19.75	**20.12**	19.19
0.1	Lena	29.82	29.81	30.31	**31.21**	29.81
	Barbara	24.17	23.95	24.50	**27.01**	23.95
	Mandrill	21.12	21.10	21.72	**22.10**	21.10
0.25	Lena	33.65	33.65	34.15	**34.88**	33.65
	Barbara	27.42	27.07	27.21	**29.10**	27.07
	Mandrill	22.89	22.80	22.93	**23.20**	22.80
0.5	Lena	36.78	36.78	36.65	**36.80**	36.78
	Barbara	**31.23**	30.84	30.01	30.30	30.84
	Mandrill	**25.09**	25.04	25.30	24.72	25.04
0.75	Lena	38.34	**38.36**	37.11	37.95	37.96
	Barbara	**33.92**	33.54	31.30	31.25	33.34
	Mandrill	**27.07**	27.01	26.80	26.12	26.01
1.00	Lena	**39.88**	39.88	38.22	38.43	39.10
	Barbara	**36.05**	35.80	34.54	32.60	35.10
	Mandrill	**28.59**	28.55	27.44	27.15	27.72
2.00	Lena	**44.11**	44.05	39.95	40.16	43.10
	Barbara	**41.90**	41.74	32.60	33.20	37.27
	Mandrill	**34.01**	33.97	28.12	28.57	30.80
3.00	Lena	**47.96**	47.87	40.82	41.83	45.46
	Barbara	**46.21**	46.05	33.90	33.68	38.96
	Mandrill	**39.12**	39.07	28.51	29.63	32.25
4.00	Lena	**52.30**	52.21	41.60	42.40	47.71
	Barbara	**50.40**	50.28	35.01	34.52	40.10
	Mandrill	**44.57**	44.53	29.32	30.26	34.53

Note: Bold values indicate the best result.

Table 6 Structural similarity index measure.

Bit rate (bpp)	Image	Modified SPIHT	SPIHT	SPHFB	LMSPIHT	Fang et al.[22]
0.01	Lena	0.6264	0.6315	0.6832	**0.7169**	0.6382
	Barbara	0.5150	0.5067	0.5510	**0.6460**	0.5082
	Mandrill	0.3626	0.3611	0.3652	**0.3874**	0.3645
0.1	Lena	0.8933	0.8929	0.8930	**0.8990**	0.8931
	Barbara	0.7974	0.7868	0.7910	**0.8019**	0.7870
	Mandrill	0.6337	0.6239	0.6356	**0.6385**	0.6240
0.25	Lena	0.9524	0.9520	0.9514	**0.9530**	0.9512
	Barbara	0.8941	0.8845	0.8921	**0.8952**	0.8883
	Mandrill	0.7736	0.7696	0.7730	**0.7749**	0.7701
0.5	Lena	0.9753	**0.9756**	0.9720	0.9750	0.9722
	Barbara	**0.9516**	0.9474	0.9480	0.9492	0.0000
	Mandrill	**0.8781**	0.8687	0.8605	0.8690	0.8713
1.0	Lena	0.9877	**0.9881**	0.9801	0.9809	0.9860
	Barbara	**0.9829**	0.9815	0.9733	0.9755	0.9790
	Mandrill	**0.9428**	0.9407	0.9121	0.9352	0.9221

Note: Bold values indicate the best result.

coefficients are found sooner than in the other methods. The $3L + 1$ thresholds corresponding to each subband are sent to the decoder in an orderly manner as part of the initial header and maintained in a short register during the entire process, to populate the LIS with sets that are inside the significant subbands. The modified SPIHT preserves the visual quality and PSNR yielded by the SPIHT for all bit rates.

References

1. M. Antonini et al., "Image coding using the wavelet transform," *IEEE Trans. Image Process.* **1**, 205–220 (1992).
2. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.* **41**, 3445–3462 (1993).
3. S. Grgic, M. Grgic, and B. Zovko-Cihlar, "Performance analysis of image compression using wavelets," *IEEE Trans. Ind. Electron.* **48**, 682–695 (2001).
4. S. Grgic, K. Kers, and M. Grgic, "Image compression using wavelets," in *Proc. IEEE Int. Symp. on Industrial Electronics*, pp. 99–104 (1999).
5. D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression: Fundamentals, Standards and Practice*, Kluwer Academic Publishers, Boston, Massachusetts (2002).
6. S. Yang et al., "Evolutionary clustering based vector quantization and SPIHT coding for image compression," *Elsevier Pattern Recognit. Lett.* **31**, 1773–1780 (2010).
7. T. Lee and H. Shen, "Efficient local statistical analysis via integral histograms with discrete wavelet transform," *IEEE Trans. Vis. Comput. Graph.* **19**(12), 2693–2702 (2013).
8. E. Cavero et al., "SPIHT-based echocardiogram compression: clinical evaluation and recommendations of use," *IEEE J. Biomed. Health Inform.* **17**(1), 103–112 (2013).
9. Z. Xinjun and W. Xingyuan, "Chaos-based partial encryption of SPIHT coded color images," *Elsevier J. Signal Process.* **93**, 2422–2431 (2013).
10. J. Ma, J. Fei, and D. Chen, "Rate-distortion weighted SPIHT algorithm for interferometer data processing," *J. Syst. Eng. Electron.* **22**, 547–556 (2011).
11. E. Christophe, C. Mailhes, and P. Duhamel, "Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3-D wavelet coding," *IEEE Trans. Image Process.* **17**, 2334–2346 (2008).

12. M. S. Manikandan and S. Dandapat, "Effective quality-controlled SPIHT-based ECG coding strategy under noise environments," *Electron. Lett.* **44**, 1182–1183 (2008).
13. W. A. Pearlman et al., "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Trans. CSVT* **14**, 1219–1235 (2004).
14. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. CSVT* **6**, 243–250 (1996).
15. C. Yushin and W. A. Pearlman, "Hierarchical dynamic range coding of wavelet subbands for fast and efficient image decompression," *IEEE Trans. Image Process.* **16**, 2005–2015 (2007).
16. S. Chang and L. Carin, "A modified SPIHT algorithm for image coding with a joint MSE and classification distortion measure," *IEEE Trans. Image Process.* **15**, 713–725 (2006).
17. H. Ochoa et al., "Set Partitioning in hierarchical frequency bands (SPHFB)," in *Proc. of IEEE Data Compression Conf.*, p. **463** (2009).
18. J. Zhu, R. M. Dansereau, and A. Cuhadar, "Error-resilient and error concealment 3-D SPIHT video coding with added redundancy," *Lec. Notes Comput. Sci.* **6134**, 351–358 (2010).
19. F. Khelifi, A. Bouridane, and F. Kurugollu, "On the SPIHT based multispectral image compression," in *Proc. IEEE Int. Symp. on Signal Processing and Information Technology*, pp. 359–363 (2006).
20. G. Higgins et al., "Lossy compression of EEG signals using SPIHT," *IET Electron. Lett.* **47**, 1017–1018 (2011).
21. H. Pan, W. Siua, and N. Lawa, "A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT," *J. Signal Process.: Image Commun.* **23**, 146–161 (2008).
22. Z. Fang et al., "Interpolation-based direction-adaptive lifting DWT and modified SPIHT for image compression in multimedia communications," *IEEE Syst. J.* **5**(4), 584–593 (2011).
23. Y. Jin, Y. Lee, and H.-J. Lee, "A new frame memory compression algorithm with DPCM and VLC in a 4 × 4 block," *EURASIP J. Adv. Signal Process.* **2009**(1), 1–18 (2009).
24. Y. Jin, Y. Lee, and H.-J. Lee, "A block-based pass-parallel SPIHT algorithm," *IEEE Trans. Circuits Syst. Video Technol.* **22**, 1064–1075 (2012).
25. J. Zhu and R. Dansereau, "Error-resilient and error concealment 3-D SPIHT for multiple description video coding with added redundancy," *IEEE Trans. Circuits Syst. Video Technol.* **22**, 855–868 (2012).
26. T. Xiang, J. Qu, and D. Xiao, "Joint SPIHT compression and selective encryption," *Appl. Soft Comput.* **21**, 159–170 (2014).
27. O. Rubio, A. Alesanco, and J. Garcia, "Secure information embedding into 1D biomedical signals based on SPIHT," *J. Biomed. Inform.* **46**, 653–664 (2013).
28. S. Isaa et al., "The effect of electrocardiogram signal compression using beat reordering and SPIHT on automatic sleep stage classification," *Eng. Procedia* **41**, 888–896 (2012).
29. A. Nandi and R. Banakar, "Throughput efficient parallel implementation of SPIHT algorithm," in *Proc. of Int. Conf. on Very Large Scale Integrated Design*, pp. 718–725 (2008).
30. N. Zhang et al., "Image compression algorithm of high-speed SPIHT for aerial applications," in *Proc. of IEEE Int. Conf. on Communication Software and Networks*, pp. 536–540 (2011).
31. R. Sudhakar and S. Jayaraman, "A new video coder using multiwavelets," in *Proc. of Int. Conf. on Signal Processing, Communications and Networking*, pp. 259–264 (2007).
32. "The USC—SIPI image database," http://sipi.usc.edu/database/ (February 2013).
33. A. Cohen, I. Daubechies, and J. C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.* **45**, 485–560 (1992).
34. A. Said and W. A. Pearlman, "SPIHT program," http://www.cipr.rpi.edu/research/SPIHT/spiht3.html#mat-spiht (January 2013).
35. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**, 600–612 (2004).

Humberto de J. Ochoa Domínguez received his BS degree in industrial electronics from the Technological Institute of Veracruz, México, his MSc degree in electronics from the Technological Institute of Chihuahua, México, and PhD degree in electrical engineering from the University of Texas at Arlington, USA. He is a full-time professor at the Universidad Autónoma de Ciudad Juárez. His current teaching and research interests include image and video coding, statistical shape analysis, and pattern recognition.

Osslan O. Vergara Villegas earned his BS degree in computer engineering from the Instituto Tecnológico de Zacatepec in 2000, his MSc degree in computer science from Center of Research and Technological Development (CENIDET) in 2003, and his PhD degree in computer science from CENIDET in 2006. He is a full-time professor at the Universidad Autónoma de Ciudad Juárez. He is a senior member of the IEEE. His fields of interest include computer vision, digital image processing, and augmented reality.

Vianey G. Cruz Sanchez earned her BS degree in computer engineering from the Instituto Tecnológico de Cerro Azul in 2000, her MSc degree in computer science from CENIDET in 2004, and her PhD degree in computer science from CENIDET in 2010. She is a full-time professor at the Universidad Autónoma de Ciudad Juárez. Her fields of interest include neurosymbolic hybrid systems, digital image processing, artificial neural networks, and augmented reality.